On learning the structure of Bayesian Networks and submodular function maximization

Giulio Caravagna*1, Daniele Ramazzotti*2, and Guido Sanguinetti1

1University of Edinburgh, Edinburgh, UK
2Stanford University, Stanford, USA

Abstract

Learning the structure of dependencies among multiple random variables is a problem of considerable theoretical and practical interest. In practice, score optimisation with multiple restarts provides a practical and surprisingly successful solution, yet the conditions under which this may be a well founded strategy are poorly understood. In this paper, we prove that the problem of identifying the structure of a Bayesian Network via regularised score optimisation can be recast, in expectation, as a submodular optimisation problem, thus guaranteeing optimality with high probability. This result both explains the practical success of optimisation heuristics, and suggests a way to improve on such algorithms by artificially simulating multiple data sets via a bootstrap procedure. We show on several synthetic data sets that the resulting algorithm yields better recovery performance than the state of the art, and illustrate in a real cancer genomic study how such an approach can lead to valuable practical insights.

1 Introduction

Learning statistical structures from multiple joint observations is a crucial problem in statistics and data science. Bayesian Networks (BNs) provide an elegant and effective way of depicting such dependencies by using a graphical encoding of conditional independencies within a set of random variables [1]. This enables a compact and intuitive modelling framework which is both highly explanatory and predictive, and justifies the enduring popularity of BNs in many fields of application [2].

Despite the undoubtable success of BNs, identifying the graphical structure underpinning a BN from data remains a challenging problem [3]. The number of possible graphs scales super-exponentially with the number of nodes [4], effectively ruling out direct search for BNs with more than a handful of nodes. Markov equivalence, the phenomenon by which two distinct graphs can encode identical conditional independence structures [5], necessarily leads to a multimodal objective function, which can be highly problematic for maximum likelihood (ML) optimisation-based and Bayesian methods alike. In practice, reasonable performance can be achieved by greedy methods that search models by their likelihood adjusted for a complexity term [6]. For information-theoretic
scoring functions, common approaches are either the Bayesian Information Criterion (BIC) by Schwarz or the Akaike Information Criterion (AIC) by Akaike \[7,8\]. For Bayesian scoring functions, popular choices are the Bayesian Dirichlet likelihood-equivalence score (BDE) \[9\] which combines the multinomial distribution with the Dirichlet prior for discrete-valued networks, or the Bayesian Gaussian equivalent (BGE) \[10\], which combines the linear Gaussian distribution with the normal-Wishart prior for Gaussian-valued networks, or the $K2$ score (K2) \[9\], another particular case of the Bayesian Dirichlet score. All of these approaches select network structures by a greedy optimisation process, either through (regularised) optimisation of the joint parameter/structure likelihood, or by optimising a collapsed likelihood where the explicit dependence on the conditional parameters is marginalised under a conjugate prior distribution. As with many non-convex optimisation problems, a schema with multiple initial conditions is often used to sample different solutions from the multimodal fitness landscape. Nevertheless, the conditions under which they should return optimal structures are poorly understood.

This paper frames the optimization problem within submodular set functions theory \[11\], a class of functions in discrete optimization problems characterized by diminishing returns, and for which greedy algorithms yield provably optimal results with high probability \[12\]. The main theoretical contribution is to show that, asymptotically, the BN score function is in expectation submodular, regardless of the regularization term. This result not only explains the practical success of greedy regularised search algorithms, but also suggests a more robust algorithm where asymptotic conditions are simulated via a bootstrap procedure \[13\]. This approach of course does not solve the Markov equivalence problem; to address that, we follow an early intuition of Koller and Friedman and devise a data-driven strategy (again based on bootstrap) to estimate a partial ordering on the set of nodes, effectively playing the role of an informative prior over graph structures \[14,15\]. Our approach therefore decouples the tasks of restricting the search space to a suitable basin of attraction, and optimising within that basin. Extensive experimentation on simulated data sets shows that the proposed algorithm outperforms several variants of regularised scores, and an experiment on a cancer genomics application shows how the approach can lead to insightful structure discovery on real life data science problems.

The rest of the paper is organised as follows. In Section 2 we introduce the necessary background, and in Section 3 we use an example to outline the intuition that motivates this work. In Sections 4 and 5 we first investigate the relation of the optimization problem within submodular set functions, and then derive a new algorithm that exploits such a relation. We conclude with case studies and discussion in Sections 6 and 7.

2 Background

In this paper we will adopt the following notation. With $D \in \mathbb{B}^{n \times m}$ we denote the input data matrix with $n$ variables and $m$ samples. For each row a variable $x_i$ is associated, with $X = \{x_1, \ldots, x_n\}$. Domain $\mathbb{B}$ can be either continuous ($\mathbb{R}$, in which case we assume to be working with Gaussian conditionals) or discrete multivariate ($\mathbb{Z}$). We aim at computing a factorization of $p(x_1, \ldots, x_n)$ from $D$. We will make use of non-parametric bootstrap techniques \[13\]: with $D \sim_k (D_1, \ldots, D_k)$ we denote $k$ non-parametric bootstrap replicates $D_i \in \mathbb{B}^{n \times m}$ of the input data $D$.

We are interested in a Bayesian Network (BN, $\mathcal{M} = \langle E, \theta \rangle$) over variables $X$, with edges $E \subseteq X \times X$ and real-valued parameters $\theta$. $E$ induces an acyclic graph over $X$, that represents
factorization

\[ p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i | \pi_i) \quad p(x_i | \pi_i) = \theta_{x_i|\pi_i}. \]

where \( \pi_i = \{x_j \mid x_j \to x_i \in E \} \) are \( x_i \)'s parents, and \( \theta_{x_i|\pi(x_i)} \) is a probability density function. The BN log-likelihood of \( M \) is given by

\[ \text{LL}(D \mid M) = \log p(D \mid E, \theta). \]

The model selection task \( D \to_{k, \Pi} M_* \), is to compute a BN \( M_* = \langle E_*, \theta_* \rangle \) by solving

\[ M_* = \arg \max_{M=(E \subseteq \Pi, \theta)} \text{LL}(D \mid M) - f(M, D) \]

where \( f \) is a regularization score \(^2\) (e.g., BIC, AIC, BDE, BGE, K2, etc.). This problem is NP-hard and, in general, one can compute a (local) optimal solution to it \(^3\). In our definition the search-space is constrained by \( E \subseteq \Pi \). Without loss of generality, we assume \( M_* \) to be estimated by a hill-climbing procedure that starts from \( k \) random initial BNs, and returns the highest scoring model. When one uses information-theoretic scoring functions, parameters are maximum-likelihood estimates (MLE) of the conditional distributions\(^2\).

We will make use also of direct acyclic graphs (DAGs), whose definition is standard; \( w_E(x_i \to x_j) \) will be the weight associated to edge \( x_i \to x_j \) in a graph with edges \( E \) via function \( w : E \to \mathbb{R} \).

**Baseline approach.** In what follows we will aim at improving over the baseline approach, which we consider to be the \( f \)-regularized selection with unconstrained search space and \( k \) initial conditions

\[ D \to_{k, 0} M_* . \]

This procedure is greedy, it starts from an initial condition \( M_0 \) — e.g., a random DAG — and performs a one-edge change (deletion or insertion of an edge) to exhaustively compute the neighbourhood \( M_0 \) of \( M_0 \). Then, \( \hat{M} \in M_0 \) is the new best solution if it has score — according to equation \(^3\) — higher than \( M_0 \) and is the maximum-scoring model in the whole neighbourhood. The greedy search then proceed recursively to examine \( \hat{M} \)'s neighbourhood, and stops if the current solution is the highest scoring in all of its neighbourhood. Thus, this search scans a set of solutions \( \{M_i\}_I \) by maximising the discrete gradient defined as

\[ \nabla_{M_i, \hat{M}} = f(\hat{M}) - f(M_i), \quad \hat{M} \in M_i \]

where \( f(M) = \text{LL}(D \mid M) - f(M, D) \) is the scoring function in equation \(^3\).

Hill Climbing is known to be suboptimal, and can be improved in several ways. For instance, instead of sampling \( k \) uncorrelated initial conditions, one can sample a model in the neighbourhood of the last computed solution (random restarts). Otherwise, one can take into account structural-equivalence classes, node orderings, and edge reversal moves to navigate the search space; see \[^{16-18}\]

\(^1\)If \( M_* \) is categorical with \( w \) values, then the multinomial estimate is

\[ \theta^\text{ML}_{x_i=x|\pi(x)=y} = \frac{n(x, y)}{\sum_{x_i=v_1} n(v_i, y)} , \]

where \( n(x_i, y) \) counts, from \( D \), the number of observed instances for an assignment of \( x_i \) and \( y \).
and references therein. Nevertheless, the number of valid solutions remains still potentially huge. For simplicity, here we consider the baseline Hill Climbing; it would be straightforward to improve our approach by adopting other search or restart strategies proposed in the literature.

In this paper we consider several common scores for BNs: the BIC, AIC, BDE, BGE and K2. In the Main Text, we discuss results obtained with the information-theoretic scores $f \in \{\text{BIC, AIC}\}$; BIC is derived as the infinite samples approximation to the MLE of the structure and the parameters of the model, and is consistent, while AIC is not. In the Supplementary Material, we present that analogous results hold for Bayesian scoring functions ($f \in \{\text{BDE, BGE, K2}\}$).
Figure 2: The 13 optima of the fitness landscape shown in Figure 1 with their BIC score. Notice the equivalence classes (discussed in Section 4.1) and the presence of optima with equivalent score but different structure. The true, i.e., generative, model is not the highest ranked in $\mathcal{F}$. If we create as poset $\Pi$ the transitive closure of the true model, however, we observe that the landscape reduces to having a unique global optima. In fact, all the optima but the true one have at least one edge not included in $\Pi$. For this $\Pi$, the landscape happens to be unimodal with a maximum at the true model; an experiment with 100 random networks shows that this happens with high probability.

### 3 An example

We begin with an example that inspired the approach that we introduce in Section 5. We will use standard terminology from the theory of BNs and optimization; formal definitions of the concepts mentioned here appear in Section 4.

Let us consider a random BN $M$ with $n = 4$ discrete nodes $\{x_1, \ldots, x_4\}$, $|\mathbb{B}| = 2$, and random conditional distributions $\theta$ (parameters). Despite being small, models of this size show a rich optimization’s landscape and allow for some visualization. In fact, the number of DAGs with $n$ nodes is super-exponential in $n$, which in this case leads to 543 models. From $M$, we generate $m = 10000$ samples and investigate the problem of identifying $M$ from such data.

With such a small network we can exhaustively construct the fitness landscape $\mathcal{F}$ of the discrete optimization, and visualize the gradient in equation (4) used to solve equation (3). The whole landscape of the Hill Climbing with BIC scores is shown in Figure 1 and shows that:

(i) there are several models with different structure but equivalent BIC score;

(ii) $M$’s BIC score is not the highest in this landscape, which has 13 optima;

(iii) the basins of attractions can be fairly large, compared to $M$’s one;

---

$G(n) = \sum_{k=1}^{n} (-1)^{k+1} (\frac{1}{2})^{k(n-k)} G(n-k)$. 

---

2 Precisely, it is computable as $G(n) = \sum_{k=1}^{n} (-1)^{k+1} (\frac{1}{2})^{k(n-k)} G(n-k)$. 

---
We expect the landscape to have multiple modes because of Markov equivalence classes (Definition 4.2), and because we are working with finite \( m \). Thus, a search in this landscape could likely be trapped in optima that are not \( M \).

We now focus on the intuition that searching for the model is generally easier if one constrains the parent sets \( \Pi \). This is often done by either setting a cutoff on \( |\pi_i| \), i.e., limiting the number of \( x_i \)'s parents, or by specifying a partially ordered set (poset) \( \Pi \subseteq \mathcal{X} \times \mathcal{X} \) such that \( x_j \) can be one of \( x_i \)'s parents only if \( (x_j, x_i) \in \Pi \). Whatever the case, the algorithmic motivation seems obvious as we drop the search’s combinatorial complexity by pruning possible solutions. However, we are interested in investigating how this affects the shape of the landscape \( F \).

We consider the constraint to be given as a poset \( \Pi \) (that, in practice, one has to estimate from data). The search is then limited to analyzing edges in \( \Pi \), so \( \Pi \) is good if it shrinks the search to visit solutions that are “closer” to \( M \) – thus, \( \Pi \) has to include \( M \)'s edges. In this example we create \( \Pi \) by adding to \( M \) also its transitive edges. In Figure 2 we show that all the models (but \( M \)) that are optima in \( F \) have at least one edge that is not allowed by \( \Pi \). So, they would not be visited by a search constrained by this \( \Pi \).

We compute the fitness landscape under \( \Pi \), \( F_{\Pi} \), and find it to have a unique optimum (Figure 2). For this poset, \( F_{\Pi} \) is unimodal with a maximum at the true model. \( M \)'s basin of attraction in \( F_{\Pi} \) is larger than in \( F \), as one might expect. This clearly suggests that we are also enjoying a simplification of the “statistical part” of the problem, which we observe with high probability (98 times out of 100) in a sample of random networks. In two cases, we observed two optima in \( F_{\Pi} \) (\( M \) and one of its subsets, data not shown). Thus, greedy optimization of equation (3) in this setting would lead to the globally optimal solution \( M \).

The above considerations are valid for the \( \Pi \) derived as transitive closure of \( M \). In real cases, of course, we do not know \( M \) and cannot trivially build this \( \Pi \). We can, however, try to approximate \( \Pi \) from \( D \). In practical cases, of course, the landscape will still be multi-modal under the approximated poset, but one would hope that the number of modes is reduced and the identification of the true model made easier in the reduced search-space.

4 Model-selection as submodular-functions optimization

Motivated by the above considerations, we investigate the relation between the fitness landscape \( F \) and a particular class of functions for discrete optimization.

4.1 Preliminaries: the fitness landscape \( F \)

Recall that \( \mathcal{M} \subseteq \mathcal{X} \times \mathcal{X} \) is the set of all possible non-reflexive edges over variables \( \mathcal{X} \).

**Definition 4.1 (Fitness).** For a subset \( \Pi \subseteq \mathcal{M} \), let \( F_{\Pi,f} : 2^\mathcal{M} \rightarrow \mathbb{R}^+ \) be the fitness function of the state space \( 2^\Pi \), data \( D \) and regularization \( f \) and the BN \( M = \langle E, \theta \rangle \) to be defined by

\[
F_{\Pi,f}(E) = \begin{cases} 
\text{LL}(D | M) - f(M, D), & \text{if } E \subseteq \Pi, \text{ } E \text{ acyclic}, \\
0, & \text{otherwise}.
\end{cases}
\]

Then, \( F_{\Pi,f}(\cdot) \) defines the fitness landscape which we use to search for a BN model \( M^{\text{MLE}} = \langle E^{\text{MLE}}, \theta^{\text{MLE}} \rangle \) that best explains \( D \) in the sense of equation (3).
So, in practice, a search that constraints the state space by $\Pi$ spans through the subspace of DAGs induced by $2^\Pi \subseteq 2^M$. Let us denote the true model as the BN $\mathbf{M}_T = (E_T, \theta_T)$, $E_T \in 2^\Pi$; for $m \to \infty$, the landscape’s MLE structure is $E_T$, when $f$ is a consistent estimator (BIC does satisfy this property, if at least one of several models contains the true distribution \cite{19}). Unfortunately, the MLE is not unique even for infinite sample size.

**Proposition 4.2** (Likelihood equivalence \cite{2}, Figure \cite{2}). For any BN $\mathbf{M} = (E, \theta)$ there exists $K_M = \{M_i = (E_i, \theta_i)\}_1$ for some index set $I$, such that $F_{M_i}(E_i) = F_{M_j}(E_j)$ for every $M_i, M_j \in K_M$.

We term $K_M$ a Markov equivalence (or I-equivalence) class of BNs with equivalent fitness value, but different structure. Thus, we can not expect to identify $M_T$ if at least one of several models contains the true distribution \cite{19}). Unfortunately, there could be models scoring higher than the ones in the landscape is multi-modal. Thus, such structures and their equivalence classes would create further optima as it happens in $\mathcal{M}_T$. For this reason, besides the problem of identifying $M_T$ within $\mathcal{M}_T$, a greedy search will likely be trapped into local optima, and heuristics use multiple initial conditions to minimize such an effect.

### 4.2 The fitness landscape $\mathcal{F}$ is a submodular function

We recall definitions from submodular set functions theory, a well-known class of functions to approximate NP-hard discrete optimization problems \cite{11}. Consider a set $\Omega$ that contains the items that can be included in the solution to our optimization problem – in this case, the edges of the model computed in equation \cite{3}. A score function over $2^\Omega$ that returns a value for each possible subset of $X \subseteq 2^\Omega$, can be used to select the solution that maximizes/ minimizes the score function.

Submodular functions are score functions with an appealing property for optimization: if $z$ is submodular over $\Omega$, the incremental value that $x$ makes when added to set $X$ – so the discrete differential $\Delta_z(\cdot)$ that we observe when we add $x$ to the solution $X$ – decreases as the size of the solution increases: $\Delta_z(X) \propto g(|X|)^{-1}$, for monotone $g$. So, for large $X$, increments due to adding elements show $\Delta_z(X \cup \{x\}) \rightarrow 0$, rendering them useless for maximization of $z$.

**Definition 4.4** (Submodular function \cite{11}). Let $z : 2^\Omega \rightarrow \mathbb{R}^+$ and $e \in \Omega$; $z$ is submodular if for every $X \subseteq Y \subseteq \Omega$ and $e \in \Omega \setminus Y$

$$z(X \cup \{e\}) - z(X) \geq z(Y \cup \{e\}) - z(Y),$$

where $\Delta_z(e \mid S)$ is $z$’s discrete derivative at $S$ with respect to $e$.

Submodularity states that, to maximize $z$, adding an element ($e \in \Omega$) to a smaller solution ($X$) helps more than adding it to a larger one ($Y \supseteq X$). Similarly to convexity and concavity in continuous optimization, submodularity allows to efficiently find provably (near-)optimal solutions.

\footnote{Indeed, Lovász has shown that for $z$ submodular its continuous limit is convex \cite{20}.}
The celebrated Nemhauser’s theorem (cfr., [12], Section 4) states in fact that the simple greedy optimization procedure

\[
X_0 = \emptyset \\
X_{i+1} = X_i \cup \{ \arg \max_{e \in \Omega} \Delta z(e \mid X_i) \}
\]

provides a good approximation to the optimal solution of the NP-hard optimization problem. In particular, for \(z\) nonnegative monotone submodular over \(\Omega\) and monotone with \(z(\emptyset) = 0\), Nemhauser has shown that for all positives integers \(k\) and \(l\),

\[
z(X_l) \geq \left( 1 - e^{-l/k} \right) \max_{|X| \leq k} z(X),
\]

and in particular, for \(l = k\), \(z(X_k) \geq (1 - 1/e) \max_{|X| \leq k} z(X)\).

We investigate the relation between such class of functions and the fitness landscape for BNs. Here, we are interested in the discrete derivative of the fitness function (Definition 4.1)

\[
\Delta F(e \mid X) = F_{M,D}(X \cup \{e\}) - F_{M,D}(X) = \sum_{y} \log \frac{p(y \mid \theta_{X \cup \{e\}}^{\text{MLE}}, X \cup \{e\})}{p(y \mid \theta_{X}^{\text{MLE}}, X)}
\]

for the MLE estimates for the parameters. Overall, for any two \(X \subseteq Y\)

\[
\Delta F(e \mid X) \geq \Delta F(e \mid Y) \iff \Delta LL(e \mid X) \geq \Delta LL(e \mid Y).
\]

when comparing the discrete derivatives of the fitness for two structures.

A novel contribution of our work is to show that, in expectation, the fitness function is submodular over the set of edges that we can include in a BN. This result does not require to use any poset \(\Pi\), and is valid on the whole state space induced by \(M\). Besides, it is a theorem on the log-likelihood function that holds also for any regularizer log-likelihood by \(\Delta LL(e \mid X) = c\) for BIC and AIC, for instance. Notice that, in the large sample size, Bayesian scores with Dirichlet priors such as BDE and BGE are equivalent to BIC [2], so that, asymptotically, such Bayesian scores will also have the general form \(\Delta LL(e \mid X)\). This leads to (for independent data \(D\))

\[
\Delta F(e \mid X) = \sum_{y} \log \frac{p(y \mid \theta_{X \cup \{e\}}^{\text{MLE}}, X \cup \{e\})}{p(y \mid \theta_{X}^{\text{MLE}}, X)}
\]

for the MLE estimates for the parameters.
for \( X \), then the Kullback-Leibler divergence between the distributions associated to \( X \) and \( X \cup \{ e \} \) is the expectation of the discrete derivative of the fitness function

\[
D_{\text{KL}}(f_{X \cup \{ e \}} \| f_X) = \mathbb{E}_{f_{X \cup \{ e \}}} [\Delta f_{X \cup \{ e \}}(e \mid X)].
\]

(10)

Then, for any valid BN structures \( X \subseteq Y \subseteq 2^\mathcal{M} \), and for every \( e \in \mathcal{M} \setminus Y \)

\[
D_{\text{KL}}(f_{X \cup \{ e \}} \| f_X) \geq D_{\text{KL}}(f_{Y \cup \{ e \}} \| f_Y)
\]

(11)

so \( f_{\mathcal{M},f} \) is in expectation a submodular function, for any consistent \( f \).

**Corollary 4.5.1** (Submodularity under the poset). For any poset \( \Pi \subseteq \mathcal{M} \), \( f_{\Pi,f} \) is in expectation a submodular function, for any consistent \( f \).

The proof of this theorem (Supplementary Materials) exploits the relation between general MLE and information theory. Intuitively, MLE can be seen as a minimization of the Kullback-Leibler divergence (where \( x \sim p \) and \( y \sim q \) are random variables)

\[
D_{\text{KL}}(x \| y) = \mathbb{E}_p \left[ \log \frac{p}{q} \right] = \int p(x) \log \frac{p(x)}{q(x)} dx
\]

among the distributions induced by the networks visited by the greedy search. In this case, since the discrete derivative of the regularizer \( f \) is a constant, the divergence is

\[
D_{\text{KL}}(f_{X \cup \{ e \}} \| f_X) = \sum_y f(y|\theta_{X \cup \{ e \}}^{\text{MLE}}, X \cup \{ e \}) \log \frac{f(y|\theta_{X \cup \{ e \}}^{\text{MLE}}, X \cup \{ e \})}{f(y|\theta_X^{\text{MLE}}, X)}.
\]

(12)

The proofs exploits the factorization of the expectation of the discrete derivative implicit in the structure of a BN, and shows that the Kullback-Leibler divergence induced by adding \( e \) to \( A \) bounds the relative divergence of \( e \) to \( B \). This result is general and one could arguably, with the same proof, show that, for any log-likelihood function over discrete sets, by standard conditions for convergence of the MLE to relative entropies, the log-likelihood is submodular in expectation.

**Corollary 4.5.1** is a trivial generalization since submodularity is closed under the subset operation.

We can also relate submodular functions to the sampling distribution.

**Corollary 4.5.2** (Submodularity under the sampling distribution). Fix finite but large enough \( m \), let \( \mathbf{D}^{(m)} \) be a sample of size \( m \) from the unknown sampling distribution \( p \) and let \( \hat{p}_m \) be the empirical estimation of the sampling distribution, under \( \mathbf{D}^{(m)} \). Then for every \( X \subseteq Y \subseteq 2^\mathcal{M} \) and \( e \in \mathcal{M} \setminus Y \)

\[
\mathbb{E}_{\hat{p}_m} \left[ D_{\text{KL}}(f_{X \cup \{ e \}} \| f_X) \geq D_{\text{KL}}(f_{Y \cup \{ e \}} \| f_Y) \right] \rightarrow_{a.s.} 1
\]

where the expectation is over the indicator function associated to inequality \( \Pi \).

The previous results allow us to draw the following conclusions. For large sample size – ideally, for \( m \to \infty \) – when the MLE estimators approach the true values, we can select a “good” model by a simple greedy search. **Corollary 4.5.2** suggests that \( f_{\mathcal{M},f} \) will be submodular in expectation also when one works with infinite bootstrap resamples computed from a single, large enough\footnote{In practice, however, Corollary 4.5.2 does not say how large the dataset has to be, or do not suggest the number of replicates that we might have to draw for practical purposes. The only thing that we know by the central limit theorem is that, pointwise, the empirical estimation of the sampling distribution \( \hat{p} \) has asymptotically normal distribution with standard rate of convergence \( \sqrt{m} \). These are inherent difficulties of bootstrap-based approaches, and we will rely on gold standards.} dataset \( \mathbf{D} \).
Algorithm 1 – Model selection for BNs via the bootstrap (Figure 3)
Steps marked with (*) can be implemented in different ways (see Sections 4.1–4.5.1).

Require: a dataset \( D \) over variables \( X \), and two integers \( k_p, k_b \gg 1 \);
1: let \( D \sim k_p \langle D_1, \ldots, D_{k_p} \rangle \) and \( M \subset X \times X \) be the set of non-reflexive edges over \( X \).
2: compute the weighted consensus structure \( \Pi_{\text{boot}} \)
\[
\Pi_{\text{boot}} = \bigcup_{i=1}^{k_p} \left\{ E_i \mid D_i \rightarrow \Pi_i, M_i = \langle E_i, \theta_i \rangle \right\}
\]
\[
w_{\Pi_{\text{boot}}} (x_i \rightarrow x_j) = \sum_{w=1}^{k_p} 1_{E_w}(x_i \rightarrow x_j); \tag{13}
\]
3: (*) remove loops from \( \Pi_{\text{boot}} \) by solving
\[
\Pi = \arg \max_{\Pi \subseteq \Pi_{\text{boot}}, \Pi \text{ acyclic}} \sum_{x_i \rightarrow x_j \in \Pi} w_{\Pi} (x_i \rightarrow x_j); \tag{14}
\]
4: let \( D \sim k_b \langle D_1, \ldots, D_{k_b} \rangle \), for any \( D_i \) generate \( \tilde{D}_i = \text{perm}(D_i) \);
5: compute \( 2k_b \) BNs under \( \Pi \)
\[
\Gamma = \{ E_i \mid D_i \rightarrow \uparrow_{\Pi}, M_i = \langle E_i, \theta_i \rangle \}, \quad \Gamma_{\text{null}} = \{ E_i \mid \tilde{D}_i \rightarrow \uparrow_{\Pi}, M_i = \langle E_i, \theta_i \rangle \}; \tag{15}
\]
6: let \( \sigma_{i,j} = [\cdots 1_x (x_i \rightarrow x_j) \cdots]_{x \in \Gamma} \) and \( \sigma_{i,j}^{\text{null}} = [\cdots 1_x (x_i \rightarrow x_j) \cdots]_{x \in \Gamma_{\text{null}}} \);
7: (*) to select \( x_i \rightarrow x_j \), test \( H \) at level \( \alpha \) with Multiple Hypotheses Correction (MHC) and output the Bayesian Network \( M = \langle E, \theta_{\text{MLE}} \rangle \) where
\[
E = \{ x_i \rightarrow x_j \mid H : E[ \sigma_{i,j} ] \neq 0, E[ \sigma_{i,j}^{\text{null}} ] \}, \quad \theta_{\text{MLE}} = \arg \max_{\theta \in \Theta} \log p(D \mid E, \theta). \tag{16}
\]

We deduce that the same greedy approximation could be used to fit a model out of each resample, and then combine them. Notice that Corollary 4.5.1 ensures that if we constrain the search space by an estimated \( \Pi \), we still enjoy these properties.

5 Model selection for BNs via empirical Bayes

Motivated by the results shown in Section 4, we present here Algorithm 1 that exploits a combination of non-parametric bootstrap estimates, likelihood-fit and hypothesis testing to select a BN model. The algorithm is conceptually divided in two phases (Figure 3) that can be customized, as we discuss in the next subsections.

Phase one: \( \Pi \)'s construction. The first phase (steps 1–3), estimates, via the bootstrap, an ordering \( \Pi \) of the model’s variables that constraints the factorization in the next phase.

Initially, the union \( \Pi_{\text{boot}} \) of all the models’ structures obtained from each of \( k_p \) non-parametric bootstrap replicates is created. This structure is called consensus as it contains the union of all the models that are obtained by a standard regularized likelihood-fit procedure. Notice that each model is obtained from one initial condition, and by scanning all possible model’s edges (via \( M \)). Each models’ parameters are dropped, and \( \Pi_{\text{boot}} \) is instead augmented with the non-parametric bootstrap scores via the set indicator function \( 1_X(y) = 1 \iff y \in X \). Thus, \( w_{\Pi_{\text{boot}}} (\cdot) \) is proportional to the edges’ frequency across the models fitted from each bootstrap replicates.

The graph induced by \( \Pi_{\text{boot}} \) is generally cyclic. In step 3, we make it acyclic by selecting a suitable subset of its edges: \( \Pi \subseteq \Pi_{\text{boot}} \). It is reasonable to maximize the scores of the edges in \( \Pi \)
model selection
hill climbing with initial conditions and -regularization; is the prior (search constraint).

non-parametric bootstrap resample under the sampling distribution

Figure 3: Graphical representation of Algorithm 1. Left: first phase (construction of the poset Π). Right: second phase (construction of the test under the poset Π).

motivated by the intuition that true model edges should have higher bootstrap scores [14]. The optimization problem of equation (14) can be solved in different ways, as we discuss in Section 5.1.

Phase two: Π’s driven model construction. The second phase (steps 4-7) is the actual selection of the final output model. In principle, we could just use the standard regularized likelihood-fit procedure to select a model under Π. Preliminary tests (data not shown), however, have highlighted an intrinsic bias in the selected output model, as a function of the regularizer f. We would like to reduce to the minimal extent this effect, while enjoying the properties of f to minimize overfit. Thus, we exploit Π to create an edge-specific statistical test to detect true edges, and create the final output model. Here, if Π is a good approximation to the transitive closure of the true model (such as in the example of Section 3), then it will drive the search to get better estimates for the test.

The null hypothesis H₀ is created from D, again by exploiting the bootstrap. We create (step 4) k_b bootstrap resamples of D; from each replicate we generate a permutation matrix \( \tilde{D}_i \in \mathbb{R}^{n \times m} \), with equivalent empirical marginal distributions. This is done by independently permuting D_i’s

6Precisely, we observed that if we here proceed by selecting a model via likelihood-fit, the variance in the estimated solution will be small and consistent with the choice of f – e.g., BIC would select sparser models than AIC – regardless how good is our estimate of Π at the previous step. We term this the phenomenon “intrinsic bias” of the regularizer.

7Let \( p_i(x_j) \) and \( \hat{p}_i(x_j) \) be the empirical marginals of \( x_j \) in \( D_i \) and \( \tilde{D}_i \). If \( x_j \) is discrete multivariate we require \( p_i(x_j) = \hat{p}_i(x_j) \). If \( x_j \) is continuous, we require the expectation and variance to be equivalent. The shuffling approach suggested in the text is consistent with this.
row vectors, which we do via function $\text{perm}()$. The joint distributions in each $\hat{D}_i$ are random, so for each pair $x_i, x_j$ we have a null model of their statistical independence normalized for their marginal distributions. Thus, if we fit a model on $D_i$ and $\hat{D}_i$ (step 5) we expect that an edge that represents a true dependency will tend to be more often present in $\Gamma$, rather than in $\Gamma_{\text{null}}$ (the models from the null hypothesis). This fitting constrains the search with $\Pi$ (the poset, so $E_i \subseteq \Pi$), it uses one initial condition and does not store the $2k_b$ parameters’ vectors.

Steps 6 and 7 perform multiple hypothesis testing for edges’ selection. We use the models computed in step 5 as a proxy to test for the dependencies. The vectors $\sigma_{i,j}$ and $\sigma^\text{null}_{i,j}$ store how many times $x_i \rightarrow x_j$ is detected in $\Gamma$ and $\Gamma_{\text{null}}$, respectively, so each $\sigma_{i,j}$ is a sample of a Binomial random variable over $k_b$ trials. Then, we can carry out a Binomial test (or, if $k_b$ is large, a 2-sided T-test) with confidence $\alpha$ and corrected for multiple testing. We will include every accepted edge $x_i \rightarrow x_j$ in the final output model $M$, augmented with the MLE of its parameters (estimated from the original dataset $D$). Notice that $M$ is acyclic as, by construction, $\Pi$ is acyclic.

**Complexity analysis.** Our procedure has cost dominated by the computation of the bootstrap estimates and likelihood-fits. However, we note that our algorithm allows for a simple parallel implementation that compute each estimates and its likelihood-fit. Once all estimates are computed, the cost of loop-breaking is proportional to the adopted heuristics, and the cost of multiple hypothesis testing is standard.

### 5.1 Removing loops from $\Pi_{\text{boot}}$

The problem of determining a DAG (here $\Pi$) from a directed graph with cycles (here $\Pi_{\text{boot}}$) is well-known in graph theory [21]. This problem consists in detecting a set of edges which, when removed from the input graph, leave a DAG – this set of edges is called feedback edge set.

In Algorithm 1 edges in $\Pi$ will constrain the search space, so it seems reasonable to remove as few of them as possible. Since the input graph is weighted by the non-parametric bootstrap coefficients, we can also interpret the cost of removing one edge as proportional to its weight. Thus, we need to figure out the minimum-cost edges to remove, which corresponds to the minimum feedback edge set formulation of the problem. In general, this problem is NP-hard and several approximate solutions have been devised (see, e.g., [22]).

We propose two different strategies to solve the optimization problem in equation (14) which are motivated by practical considerations.

1. **(confidence heuristic).** An approximate solution to the problem can be obtained by a greedy heuristics that breaks loops according to their weight $w_{\Pi_{\text{boot}}}$. The approach is rather intuitive: one orders all the edges in $\Pi_{\text{boot}}$ based on their weight – lower scoring edges are considered first. Edges are then scanned in order according to their score and removed if they cause any loop in $\Pi_{\text{boot}}$. This approach is, in general, sub-optimal.

The algorithmic complexity of the method depends first on sorting the edges and on the subsequent loop detection. Given a number of $a$ edges in $\Pi_{\text{boot}}$, they can be sorted with a sorting algorithm, e.g., quicksort [23], with a worst case complexity of $O(a^2)$ (average complexity for quicksort $O(a \log a)$). Then, for each ordered edge, we evaluate loops, e.g., either by depth-first search or breadth-first search (complexity $O(n + a)$, with $n$ being the number of vertices [24]). This leads to a total complexity of $O(a^2) + O(n + a)$ in the worst case for removing the loops.
2. (agony). In [25], Gupte et al. define a measure of the hierarchy existing in a directed graph. Given a directed graph \( G = (V, E) \), let us consider a ranking function \( r : V \rightarrow \mathbb{N} \) for the nodes in \( G \), such that \( r(u) < r(v) \) expresses the fact that node \( u \) is "higher" in the hierarchy than \( v \). If \( r(u) < r(v) \), then edge \( u \rightarrow v \) is expected and does not cause any "agony". On the contrary, if \( r(u) \geq r(v) \) edge \( u \rightarrow v \) would cause agony.

We here remark that any DAG induces a partial order over its nodes, and, hence, it has always zero agony: the nodes of a DAG form a perfect hierarchy. Although the number of possible rankings of a directed graph is exponential, Gupte et al. provide a polynomial-time algorithm for finding a ranking of minimum agony. In a more recent work, Tatti et al. [26] provide a fast algorithm for computing the agony of a directed graph. With \( a \) being the number of edges of \( G \), the algorithm has a theoretical bound of \( O(a^2) \) time.

Therefore, we can compute a ranking over \( \Pi_{\text{boot}} \) at minimum agony, i.e., a ranking of the nodes with small number of inconsistencies in the bootstrap resampling, thus which maximizes the overall confidence. With such a ranking, we can solve equation (14) by removing from \( \Pi_{\text{boot}} \) any edge which is inducing agony.

**Proposition 5.1.** The poset \( \Pi \) built by agony is a superset of the one computed by confidence heuristic. See Figure 3.

### 5.2 Multiple hypothesis testing

**Correction for Multiple Hypotheses Testing (MHC)** can be done in two ways: one could correct for false discovery rate (FDR, e.g., Benjamini-Hochberg) or family-wise error rate (FWER, e.g., Holm-Bonferroni). The two strategies have different motivation: FWER corrects for the probability of at least one false positive, while FDR for the proportion of false positives among the rejected null hypotheses. Thus, FWER is a stricter correction than FDR.

Given these premises, it is possible to define a rule of thumb. If one has reason to believe that \( \Pi \) is "close" to the true model, i.e., \( \Pi \) has few false positives, then a less stringent correction such as FDR could be appropriate. Otherwise, a FWER approach might be preferred.

Multiple hypotheses testing is also influenced by the number of tests that we carry out. We perform \( |\Pi| \) tests, and hence FWER scales as \( \alpha/|\Pi| \). The theoretical bound on \( |\Pi| \) is the size of the biggest direct acyclic graphs over \( n \) nodes

\[
|\Pi| \leq \left( \sum_{i=0}^{n} n - i \right) - n = \frac{n(n+1) - 2n}{2} \leq |\Pi_{\text{boot}}| = \mathcal{O}(n^2).
\]

(17)

Thus, the size of \( \Pi_{\text{boot}} \) is a bound to the number of tests. In general, because of the regularization term in the model fit of equation [13], one expects \( |\Pi_{\text{boot}}| \ll n^2 \).

### 6 Case studies

We performed extensive comparisons of our approach to the baseline Hill Climbing by generating synthetic data. Then, we tested the algorithm against a well-known BN benchmark, and against real cancer genomics data. We provide R implementation of all the methods mentioned in this manuscript, as well as sources to replicate all our findings (Supplementary Data). For Hill Climbing, we used the bnlearn package [27].
Figure 4: Performance with synthetic data for binary variables with $f = \text{BIC}$. In top panel we show precision (PPV) and recall (TPR) for BNs with $n$ nodes, density $\delta$, and $m$ samples perturbed at noise rate $\nu$. We compare Hill Climbing with $k = \{0, 200\}$ (D $\rightarrow_k M$) against Algorithm 1 with $k_p = k_b = 100$. 100 BNs for each parameter configuration are generated. The trends suggest a similar PPV but better TPR for Algorithm 1 in all settings. The performance with the confidence $\Pi$ seems independent of multiple hypotheses correction, which instead impacts on the performance with the agony $\Pi$ (FDR 0.2). Other tests carried out for $n = 10$, $\delta = \{0.4, 0.6\}$, $m = \{50, 100\}$, continuous variables and Bayesian scores confirm these trends (Supplementary Figures S8, S9, S10, and S11). In the bottom-left panel we show the density of the inferred models for different values of $\delta$, highlighting the intrinsic tendency of the plain regularization to low $\delta$. In the bottom-right panel we measure the overlap between the posets $\Pi$ built by confidence or agony, providing evidence to support Proposition 5.1.

6.1 Tests with synthetic data

We carried out an extensive performance test that we recapitulate here and in the Supplementary Material. The aim of the test is to assess which configuration of poset and hypotheses testing performs best for Algorithm 1 and compare its performance against Hill Climbing. We generated random networks (structures and parameters) with different densities – i.e., number of edges with respect to number of variables – and various number of variables. From those BNs and a random (uniform) probability associated with each edge, we generated several datasets and perturbed them with different rates of false positives and negatives (noise). For each model inferred, we computed standard scores of precision (positive predictive value, PPV) and recall (true positives rate TPR).

Results for discrete networks with the $f = \text{BIC}$ are shown in Figure 4. For continuous networks (Gaussian) with also $f = \text{AIC}$ in Supplementary Figure S8. Analogous tests for Bayesian scoring functions are in Supplementary Figures S9 ($f = \text{BDE}$), S10 ($f = \text{K2}$) and S11 ($f = \text{BGE}$).
comparison suggests that Algorithm 1 has a similar ability to retrieve true edges of Hill Climbing, PPV, but also a tendency to retrieve models with more edges, TPR. Thus, in all settings Algorithm 1 seems to improve remarkably over the baseline approach. The comparison suggests also that edge-selection by hypotheses testing seems less biased towards returning sparse models than a procedure based only on regularization. However, both approaches seem to converge towards fixed densities of the inferred model, with Algorithm 1 giving almost twice as many edges as Hill Climbing.

The effect of $k$ independent initial conditions for the Hill Climbing procedure does not seem to provide noteworthy improvements. Similarly, strategies for MHC do not seem to increase the performance in a particular way. For for agony, a stringent correction – FWER – seems too reduce TPR, while FDR does not seem to affect the scores. MHC does not seem to have any effect on the confidence poset. Interestingly, the comparison provides evidence that the agony poset is a superset of the confidence one, as the percentage of edges of the latter missing from the former approaches almost 0. Other tests with these data suggest a minor improvement of performance if we use 1000 bootstrap resamples, or different configurations of the parameters (data not shown). It is worth also to observe that, concerning the second bootstrap to create the null models, no major changes where detected for larger $k_b$; so in practice $k_b = 100$ could be considered as a suitable value across multiple application domains.

### 6.2 The alarm network

We consider the standard alarm network benchmark, as provided in the bnlearn package. alarm has $n = 37$ variables connected through 46 edges, for a total of 509 parameters.

In Figure 5 we show the result of model selection for large samples size and $f = \text{BIC}$. The comparison is performed against Hill Climbing with $k = 0$ and $k = 200$, whereas Algorithm 1 is executed with $k_p = k_b = 100$. For large $m$, most setting seem to achieve the same performance; for lower $m$, highest PPV and TPR are achieved by Algorithm 1 (confidence, FWER). For this model, the use of multiple initial conditions for the Hill Climbing procedure reduces TPR: this is due to the number of spurious edges estimated, as the number of true positives is the same for $k = 0$ and $k = 200$. The models inferred by Algorithm 1 are strictly contained, and the confidence poset has higher scores than the agony one.

For this particular network we investigated also the effect of different sample size $m$, and the p-value for the statistical test on the performance of the algorithms. In Figure 6 we show boxplots obtained from 100 datasets generated with different sample sizes. Results suggest minor changes in the performance with $m \geq 10^4$, and generalize the findings of Figure 5. Log-log plots show a consistent gap in the p-value statistics for the two models computed by Algorithm 1 shown in Figure 5. This is a phenomenon that we observed in all synthetic tests for sufficiently large $m$ (data not-shown), and that suggests the correctness of the statistical test in Algorithm 1.

Analysis of the variation of the performance as a function of the p-values’ cutoff – for $p < 0.05$, $p < 10^{-2}$ and $p < 10^{-3}$ with $m = 100$ – shows small increase in PPV for lower p-values, but not meaningful changes in TPR scores (Supplementary Figure S12).
6.3 Modeling cancer evolution from genomic data

Cancers progress by accumulating genetic mutations that allow cancer cells to grow and proliferate out of control \[31\]. Mutations occur by chance, i.e., as a random process, and are inherited through divisions of cancer cells. The subset of mutations that trigger cancer growth by allowing a clone to expand, are called drivers \[32\]. Drivers, together with epigenetic alterations, orchestrate cancer initiation and development with accumulation and activation patterns differing between individuals \[33\]. This huge genotypic diversity – termed tumor heterogeneity – is thought to lead to the emergence of drug-resistance mechanisms and failure of treatments \[34\].

Major efforts are ongoing to decipher the causes and consequences of tumor heterogeneity, and its relation to tumor progression (see, e.g., \[35\], and references therein). Here, we consider the problem of inferring a probabilistic model of cancer progression that recapitulates the temporal ordering, i.e., qualitative clocks, of the mutations that accumulate during cancer evolution \[36\]. We do this by

---

8Correlated restarts improve Hill Climbing solutions (data not shown). However, for a fair comparison with Algorithm 1 we should have then correlated the initial solutions used to compute II. To avoid including a further layer of complexity to all the procedures, we rather not do that.
scanning snapshots of cancer genomes collected via biopsy samples of several primary tumors; all the patients are untreated and diagnosed with the same cancer type (e.g., colorectal).

In this model-selection problem variables are $n$ somatic mutations detected by DNA sequencing – e.g., single-nucleotide mutations or chromosomal re-arrangements – annotated across $m$ independent samples. Thus, a sample is an $n$-dimensional binary vector: $B = \{0, 1\}$, and $x_i = 0$ if the $i$-th lesion is not detected in the patient’s cancer genome. We aim at inferring a model that accounts for the accumulation of the input variables during tumor evolution in different patients.

BNs do not encode explicitly this “cumulative” feature; however, they were recently combined with Suppes’ theory of probabilistic causation \cite{suppes1970causality}, which allows to describe cumulative phenomena. **Suppes-Bayes Causal Networks (SBCNs)** \cite{cnon2019} are BNs whose edges satisfy Suppes’ axioms for probabilistic causation, which mirror an expected “trend of selection” among the lesions, which is at the base of a Darwinian interpretation of cancer evolution \cite{finkelstein2015}. Suppes’ conditions take the form of inequalities over pairs of variables that are evaluated before model-selection via a non-parametric bootstrap procedure. The model-selection’s landscape is then pruned of the edges that do not satisfy such conditions; thus, we can frame this as a poset

$$\Pi_{\text{Suppes}} = \{x_i \rightarrow x_j | p(x_i) > p(x_j | x_i) \wedge p(x_j | x_i) > p(x_j | -x_i)\}$$  \hspace{1cm} (18)

that we estimate from $D$, along the lines of \cite{le2021}. The parameters $\theta$ of a SBCN will encode these conditions implicitly, rendering them suitable to model cumulative diseases such as cancer or HIV \cite{cnon2019}

We will use data from \cite{le2021}, which collected and pre-processed high-quality genomics profiles from The Cancer Genome Atlas (TCGA). We consider a dataset of $m = 152$ samples and $n =$ \footnote{https://cancergenome.nih.gov/}
Figure 7: We estimated a model of progression of colorectal cancer (CRC) from a set of MSS tumors studied in [29]. Before inference, a set of boolean formulae is computed and added to the input data as new variables. These represent non-linear combinations of mutations and copy numbers alterations (CNAs) in the original genes, as computed in [29]. In top, we show the graphical notation of a formula that involves the genes activating the PI3K pathway; the intuition of a formula is to capture a functional module that is disrupted by mutations/ CNAs differently across all patients. The model is then obtained with $k_p = k_b = 100$ and the same $\Pi_{\text{Suppes}}$ estimated in [29] via Wilcoxon test ($p < 0.05$), after the marginal and conditional distributions are assessed with $k_p$ bootstrap resamples. In the test construction ($p < 0.01$), we also use 100 correlated restarts of the Hill Climbing to get better estimates for $\Gamma$. The linear progression model is due to Fearon and Vogelstein [30].
54 variables, which refers to colorectal cancer patients with clinical Microsatellite Stable Status (MSS). The input data for MSS tumors consists in mutations (mut, mostly missense etc.) and copy numbers (amp, high-level amplifications; del, homozygous deletions) detected in 21 genes of 5 pathways that likely drive colon cancer progression. 20 out of 54 variables are obtained as non-linear combinations of mutations and copy numbers in the original genes. For instance,

\[ x_g = x_{PIK3CA:mut} \lor x_{IGF2:amp} \lor x_{ERBB2:amp} \lor x_{ERBB2:mut} \lor (x_{Pten:mut} \oplus x_{Pten:del}) \]

is a variable \( x_g \) associated to the combination (in disjunctive \( \lor \) and exclusive \( \oplus \) form) of the events associated to the driver genes of the PI3K pathway PIK3CA, IGF2, ERBB2 and PTEN. These new variables are called formulas (see [29] for a full list) and are included in \( D \) before assessment of Suppes’ conditions for two reasons. They capture the inter-patient heterogeneity observed across the TCGA cohort (i.e., as biological “priors”). They limit the confounding effects of attempting inferences from heterogenous populations (i.e., as statistical “priors”).

We execute only the second part of our algorithm, i.e., the test, and compare the inferred model against the one obtained by Hill Climbing constrained by \( \Pi_{\text{Suppes}} \) and with one initial condition (Figure 4 in [29]). In Figure 4 we show the model obtained with \( k_p = k_b = 100 \) and the same \( \Pi_{\text{Suppes}} \) estimated in [29] via Wilcoxon test \( (p < 0.05) \) after the marginal and conditional distributions are assessed with \( k_p \) bootstrap resamples. In the test construction \( (p < 0.01) \), we also use 100 correlated restarts of the Hill Climbing to get better estimates for \( \Gamma \).

We observe how our model is capable of capturing a lot of known features of MSS tumors as described in the seminal work of [30]. In fact, we find APC as the main gene starting the progression followed by KRAS. Afterward, we observe multiple branches, yet involving genes from the PI3K (i.e., PIK3CA) and TGFb (i.e., SMAD2 and SMAD4) pathways, which are suggested to be later events during tumorigenesis of MSS tumors. While TP53 is not inferred to be a late event in the progression, we still find the P53 pathway to be involved in advanced tumors with ATM being one of the final nodes in one branch of the model. We remark that this tumor type shows considerable heterogeneity across different patients [41], and evidences of TP53 as an early event in this cancer’s progression have been found [42].

7 Conclusions

In this paper we consider the identification of a factorization of a BN without hidden variables. This model-selection task is central to problems in statistics that require the learning of a joint distribution made compact by retaining only the relevant conditional dependencies in the data.

A common approach to it consists of a heuristic search over the space of factorizations, the result being the computation of the MLE of the structure and the parameters of the model, or of a marginalised likelihood over the structures. Surprisingly, the simple Hill-Climbing search strategy augmented with a regularized score function, provides satisfactory baseline performance [6].

Here, we give a framing of the foundations of this optimization problem within maximization of submodular set functions, and show that the fitness landscape is, in expectation, submodular. Then, we derive an algorithm based on bootstrap and multiple hypothesis testing that, compared to baseline greedy optimization, achieves consistently better model estimates. This result can

---

10 The study in [29] analyses also highly Microsatellite Instable tumors. Unfortunately, that subtype’s data are associated to a very small dataset of \( m = 27 \) samples, and thus we here focus only on Microsatellite Stable tumors, a common subtype classification of such tumors.
stimulate other approaches to exploit the intriguing relation between the log-likelihood function of a BN, and this class of optimization problems that allow for a greedy optimization. The theorem that we prove, which is grounded on the relation between MLE estimation and information theory via the Kullback-Leibler divergence, seems to suggest that this result can be extended to a broader class of likelihood functions for discrete optimization.

Besides the connection to this class of functions, this paper attempts also at unifying two streams of research in BN model-selection.

On one side, we draw inspiration from the seminal works by Friedman et al. which investigated whether we can assess “if the existence of an edge between two nodes is warranted”, or if we “can say something of the ordering of two variables” [14]. Precisely, Friedman et al. answered to these questions by showing that high-confidence estimates on certain structural features, when assessed by a non-parametric bootstrap strategy, can be “indicative of the existence of these features in the generative model”.

On the other side, we follow the suggestion by Teyssier and Koeller on the well-known fact that the best network consistent with a given node ordering can be found very efficiently [15]. Teyssier and Koeller consider BNs of bounded in-degree, and “propose a search not over the space of structures, but over the space of orderings, selecting for each ordering the best network consistent with it”. Their motivation is driven by algorithmic an argument: “[the orderings] search space is much smaller, makes more global search steps, has a lower branching factor, and avoids costly acyclicity checks”.

Here, we connect the two observations in one framework. We first estimate orderings via non-parametric bootstrap, combined with greedy estimation of the model in each resample. Then, after rendering the model acyclic, we use it to select one final model that is consistent with the orderings. Our approach improves regardless of the information-theoretic or Bayesian scoring function adopted. To this extent, we use the orderings as an empirical Bayes prior over model structures, and compute the maximum a posteriori estimate of the model. The parameters are then the MLE estimates for the selected structure. Our result is based on a refinement of the original observation by Teyssier and Koller: when we know the ordering, besides improving complexity we enjoy a systematic reduction in the “statistical” complexity in the problem of identifying true dependencies. We postulate this after observing that with the best possible ordering – i.e., a transitive closure of the generative model – the fitness landscape becomes unimodal. This is consistent with greedy optimization of submodular functions.

The asymptotic submodularity of the fitness function provides an important justification for optimisation-based structure learning methods. These are complementary to Bayesian approaches and, while optimisation methods often provide good performance with reasonable computational costs [43], Bayesian methods offer considerable advantages for uncertainty quantification and principled incorporation of prior information [11,15]. The implications of asymptotic submodularity for Bayesian methods are not clear at the moment, but it is conceivable that future research in this direction could lead to algorithmic benefits also in a Bayesian framework. More generally, optimisation methods by construction return a single, best scoring structure; while this can be a reasonable approximation in certain situations, in general conclusions based on a single optimal graph may sometimes be misleading [46]. One possible way to overcome this burden is to sample networks from the posterior distribution, see [17] for a discussion. It is an interesting question whether our data-resampling procedure may in itself be used to provide a measure of confidence in individual edges, for example by connecting bootstrap scores with (empirical) Bayesian posteriors.

The whole theory could be further improved, and compared to other well-known approaches. For
instance, one could investigate whether these results hold for different regularization strategies such as LASSO [47]. Besides, our approach might be also framed as a James-Stein alike estimator [48]. In our method, we are exploiting the observation that the joint estimation of all the possible parents at once, as given by combining the greedy fit through multiple bootstrap resample, gives better estimates than the independent estimation of each parent set. This resembles the idea of James-Stein estimator, that is that measurements should be combined if one is interested in minimizing their total error; this may allow to formally state the relation between submodular functions and shrinkage estimators.

**Acknowledgement.** G.C. and G.S. acknowledge support from the European Research Council under grant MLCS306999. All the authors wish to thank Dirk Husmeier for useful discussions on a preliminary version of this manuscript.

**References**

[1] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Morgan Kaufmann, 1988.

[2] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques.* MIT press, 2009.

[3] David Maxwell Chickering, David Heckerman, and Christopher Meek. Large-sample learning of bayesian networks is np-hard. *The Journal of Machine Learning Research*, 5:1287–1330, 2004.

[4] Robert W Robinson. Counting unlabeled acyclic digraphs. In *Combinatorial mathematics V*, pages 28–43. Springer, 1977.

[5] Judea Pearl and Thomas S Verma. A theory of inferred causation. *Studies in Logic and the Foundations of Mathematics*, 134:789–811, 1995.

[6] José A Gámez, Juan L Mateo, and José M Puerta. Learning bayesian networks by hill climbing: efficient methods based on progressive restriction of the neighborhood. *Data Mining and Knowledge Discovery*, 22(1):106–148, 2011.

[7] Gideon Schwarz. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

[8] Hirotogu Akaike. Information theory and an extension of the maximum likelihood principle. In *Selected Papers of Hirotogu Akaike*, pages 199–213. Springer, 1998.

[9] Gregory F Cooper and Edward Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine learning*, 9(4):309–347, 1992.

[10] Dan Geiger and David Heckerman. Learning gaussian networks. In *Proceedings of the Tenth international conference on Uncertainty in artificial intelligence*, pages 235–243. Morgan Kaufmann Publishers Inc., 1994.
[11] Andreas Krause and Daniel Golovin. Submodular function maximization. *Tractability: Practical Approaches to Hard Problems*, 3(19):8, 2012.

[12] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294, 1978.

[13] Bradley Efron and Robert J Tibshirani. *An Introduction to the Bootstrap*. CRC press, 1994.

[14] Nir Friedman, Moises Goldszmidt, and Abraham Wyner. Data analysis with bayesian networks: A bootstrap approach. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 196–205. Morgan Kaufmann Publishers Inc., 1999.

[15] Marc Teyssier and Daphne Koller. Ordering-based search: A simple and effective algorithm for learning bayesian networks. *arXiv preprint arXiv:1207.1429*, 2012.

[16] David Maxwell Chickering. Learning equivalence classes of bayesian-network structures. *Journal of machine learning research*, 2(Feb):445–498, 2002.

[17] Nir Friedman and Daphne Koller. Being bayesian about network structure. a bayesian approach to structure discovery in bayesian networks. *Machine learning*, 50(1-2):95–125, 2003.

[18] Marco Grzegorczyk and Dirk Husmeier. Improving the structure mcmc sampler for bayesian networks by introducing a new edge reversal move. *Machine Learning*, 71(2):265–305, 2008.

[19] Dominique MA Haughton et al. On the choice of a model to fit data from an exponential family. *The Annals of Statistics*, 16(1):342–355, 1988.

[20] László Lovász. Submodular functions and convexity. In *Mathematical Programming The State of the Art*, pages 235–257. Springer, 1983.

[21] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[22] Viggo Kann. *On the approximability of NP-complete optimization problems*. PhD thesis, Royal Institute of Technology Stockholm, 1992.

[23] Charles AR Hoare. Quicksort. *The Computer Journal*, 5(1):10–16, 1962.

[24] Thomas H Cormen. *Introduction to algorithms*. MIT press, 2009.

[25] Mangesh Gupte, Pravin Shankar, Jing Li, Shanmugueelayut Muthukrishnan, and Liviu Iftode. Finding hierarchy in directed online social networks. In *Proceedings of the 20th international conference on World wide web*, pages 557–566. ACM, 2011.

[26] Nikolaj Tatti. Hierarchies in directed networks. In *Data Mining (ICDM), 2015 IEEE International Conference on*, pages 991–996. IEEE, 2015.

[27] Marco Scutari. Learning bayesian networks with the bnlearn r package. *Journal of Statistical Software*, 35(03), 2010.
[28] Ingo A Beinlich, Henri J Suermondt, R Martin Chavez, and Gregory F Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. Springer, 1989.

[29] Giulio Caravagna, Alex Graudenzi, Daniele Ramazzotti, Rebeca Sanz-Pamplona, Luca De Sano, Giancarlo Mauri, Victor Moreno, Marco Antoniotti, and Bud Mishra. Algorithmic methods to infer the evolutionary trajectories in cancer progression. Proceedings of the National Academy of Sciences, 113(28):E4025–E4034, 2016.

[30] Eric R Fearon, Bert Vogelstein, et al. A genetic model for colorectal tumorigenesis. Cell, 61(5):759–767, 1990.

[31] Peter C Nowell. The clonal evolution of tumor cell populations. Science, 194(4260):23–28, 1976.

[32] Bert Vogelstein and Kenneth W Kinzler. Cancer genes and the pathways they control. Nature medicine, 10(8):789–799, 2004.

[33] Franziska Michor, Yoh Iwasa, and Martin A Nowak. Dynamics of cancer progression. Nature reviews cancer, 4(3):197–205, 2004.

[34] Charles Swanton. Intratumor heterogeneity: evolution through space and time. Cancer research, 72(19):4875–4882, 2012.

[35] Andriy Marusyk and Kornelia Polyak. Tumor heterogeneity: causes and consequences. Biochimica et Biophysica Acta (BBA)-Reviews on Cancer, 1805(1):105–117, 2010.

[36] Niko Beerenwinkel, Roland F Schwarz, Moritz Gerstung, and Florian Markowetz. Cancer evolution: mathematical models and computational inference. Systematic biology, 64(1):e1–e25, 2015.

[37] Patrick Suppes. A probabilistic theory of causation, 1970.

[38] Daniele Ramazzotti, Alex Graudenzi, Giulio Caravagna, and Marco Antoniotti. Modeling cumulative biological phenomena with suppes-bayes causal networks, 2017.

[39] Daniele Ramazzotti, Giulio Caravagna, Loes Olde Loohuis, Alex Graudenzi, Ilya Korsunsky, Giancarlo Mauri, Marco Antoniotti, and Bud Mishra. Capri: efficient inference of cancer progression models from cross-sectional data. Bioinformatics, 31(18):3016–3026, 2015.

[40] The Cancer Genome Atlas Network et al. Comprehensive molecular characterization of human colon and rectal cancer. Nature, 487(7407):330–337, 2012.

[41] Justin Guinney, Rodrigo Dienstmann, Xin Wang, Aurélien de Reyniès, Andreas Schlicker, Charlotte Soneson, Laetitia Marisa, Paul Roepman, Gift Nyamundanda, Paolo Angelino, et al. The consensus molecular subtypes of colorectal cancer. Nature medicine, in print, 2015.

[42] Noa Rivlin, Ran Brosh, Moshe Oren, and Varda Rotter. Mutations in the p53 tumor suppressor gene important milestones at the various steps of tumorigenesis. Genes & cancer, 2(4):466–474, 2011.
[43] Matthieu Vignes, Jimmy Vandel, David Allouche, Nidal Ramadan-Alban, Christine Cierco-Ayrolles, Thomas Schiex, Brigitte Mangin, and Simon De Givry. Gene regulatory network reconstruction using bayesian networks, the dantzig selector, the lasso and their meta-analysis. PloS one, 6(12):e29165, 2011.

[44] Dirk Husmeier. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic bayesian networks. Bioinformatics, 19(17):2271–2282, 2003.

[45] Sach Mukherjee and Terence P Speed. Network inference using informative priors. Proceedings of the National Academy of Sciences, 105(38):14313–14318, 2008.

[46] Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe’er. Using bayesian networks to analyze expression data. Journal of computational biology, 7(3-4):601–620, 2000.

[47] Robert Tibshirani. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological), pages 267–288, 1996.

[48] William James and Charles Stein. Estimation with quadratic loss. In Proceedings of the fourth Berkeley symposium on mathematical statistics and probability, volume 1, pages 361–379, 1961.

A Supplementary Figures

The following figures are provided.

- Figure S8: synthetic tests with different settings from Figure 4.
- Figure S9, S10 and S11: synthetic tests analogous to the ones from Figure 4 for Bayesian scoring functions.
- Figure S12: the effects of different p-values on the model-selection for the alarm network.

B Proofs

B.1 Auxiliary lemma

Consider the entropy $H : 2^X \to [0, \infty)$, the cross-entropy $H(\cdot, \cdot)$ and the conditional entropy $H(\cdot \mid \cdot)$ as usual.

**Lemma B.1.** $D_{KL}(p(x \mid zy)\|q(x \mid y)) = D_{KL}(p(xyz)\|q(xy)) + H(y) - H(z, y) - H(z\mid y)$

24
Proof.

\[ D_{\text{KL}}(p(xyz)\|q(xy)) = \sum_{x,y,z} p(xyz) \log \frac{p(xyz)}{q(xy)} \]
\[ = \sum_{x,y,z} p(xyz) \log \frac{p(x)}{q(y)} \cdot \frac{p(z \mid xy)}{q(x \mid y)} \]
\[ = \sum_{x,y,z} p(xyz) \log \frac{p(x)}{q(y)} + \sum_{x,y,z} p(xyz) \log \frac{p(z \mid xy)}{q(x \mid y)} \]

Rewrite

\[ [\star] = \sum_{x,y} p(xy) \log \frac{p(xy)}{q(y)} \quad \text{(} \sum_z p(xyz) = p(xy) \text{)} \]
\[ = D_{\text{KL}}(p(xy)\|q(y)) \]
\[ = \sum_{x,y} p(xy) \log p(xy) - \sum_{y} \sum_{x} p(xy) \log q(y) = \]
\[ = \sum_{x,y} p(xy) \log p(xy) - \sum_{y} p(y) \log q(y) = \quad \text{(} \sum_x p(xy) = p(y) \text{)} \]
\[ = H(x, y) - H(y) \]
\[ \text{H(x, y) - H(y)} \]

\[ [\star\star] = \sum_{x,y,z} p(xyz) \log \frac{p(x \mid zy)p(zy)/p(xy)}{q(x \mid y)} \quad \text{(} \sum_z p(xyz) = p(xy) \text{)} \]
\[ = \sum_{x,y,z} p(xyz) \log \frac{p(x \mid zy)}{q(x \mid y)} + \sum_{x,y,z} p(xyz) \log \frac{p(zy)}{p(xy)} \]

It is easy to show that

\[ [\star] = \sum_{z,y} p(zy) \sum_{x} p(x \mid zy) \log \frac{p(x \mid zy)}{q(x \mid y)} \]
\[ = D_{\text{KL}}(p(x \mid zy)\|q(x \mid y)) \quad \text{(def.)} \]

\[ [\star\star] = \sum_{z,y} \sum_{x} p(xyz) \log p(zy) - \sum_{x,y} \sum_{z} p(xyz) \log p(xy) \]
\[ = \sum_{z,y} p(zy) \log p(zy) - \sum_{x,y} p(xy) \log p(xy) \quad \text{(} \sum_z p(xyz) = p(zy), \sum_z p(xyz) = p(xy) \text{)} \]
\[ = H(z, y) - H(x, y) \]

25
Thus, by putting all together

\[
D_{KL}(p(xyz)||q(xy)) = H(x, y) - H(y) + D_{KL}(p(x | zy)||q(x | y)) + H(z, y) - H(x, y)
= D_{KL}(p(x | zy)||q(x | y)) + H(z, y) - H(y).
\]

\[\square\]

**Lemma B.2.** \(D_{KL}(p(x | zy)||q(x | y)) \geq D_{KL}(p(x | zyy')||q(x | yy'))\).

**Proof.** By Lemma[B.1]

\[
D_{KL}(p(x | zy)||q(x | y)) = D_{KL}(p(xyz)||q(xy)) - H(z | y)
D_{KL}(p(x | zyy')||q(x | yy')) = D_{KL}(p(xyy'z)||q(xyy')) - H(z | yy')
\]
so we need to show that

\[
D_{KL}(p(xyz)||q(xy)) - H(z | y) \geq D_{KL}(p(xyy'z)||q(xyy')) - H(z | yy')
\]

Let us focus on the right hand side and exploit the concavity of the logarithm function and Jensen’s inequality – if \(X\) is a random variable and \(f\) is a convex function, then \(f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)]\).

\[
D_{KL}(p(xyy'z)||q(xyy')) = \mathbb{E}_{p(xyy')} \left[ \log \frac{p(xyy')}{q(xyy')} \right]
= \mathbb{E}_{p(xyz)} \mathbb{E}_{p(y' | xyz)} \left[ \log \frac{p(xyz)p(y' | xyz)}{q(xyy')} \right]
= \mathbb{E}_{p(xyz)} \mathbb{E}_{p(y' | xyz)} \left[ -\log \frac{q(xyy')}{p(xyz)p(y' | xyz)} \right]
\leq \mathbb{E}_{p(xyz)} - \log \mathbb{E}_{p(y' | xyz)} \left[ \frac{q(xyy')}{p(xyz)p(y' | xyz)} \right] \quad \text{(by Jensen inequality)}
\]

\[
= \mathbb{E}_{p(xyz)} \left[ -\log \sum_{y'} p(y' | xyz) \frac{q(xyy')}{p(xyz)p(y' | xyz)} \right]
= \mathbb{E}_{p(xyz)} \left[ \log \frac{p(xyz)}{\sum_{y'} q(xyy')} \right]
= \mathbb{E}_{p(xyz)} \left[ \log \frac{p(xyz)}{q(xy)} \right] = D_{KL}(p(xyz)||q(xy))
\]

So we have that

\[
D_{KL}(p(xyy'z)||q(xyy')) \leq D_{KL}(p(xyz)||q(xy))
\]

Let us define \(k \geq 0\) to be the difference

\[
D_{KL}(p(xyy'z)||q(xyy')) = D_{KL}(p(xyz)||q(xy)) - k
\]
and rewrite the lemma inequality as

\[
D_{KL}(p(xyz)||q(xy)) - H(z | y) \geq D_{KL}(p(xyz)||q(xy)) - k - H(z | yy')
\]

26
which leads to
\[-H(z \mid y) \geq -k - H(z \mid yy')\]
and hence
\[k \geq H(z \mid y) - H(z \mid yy').\]
Now recall that
\[
H(z \mid y) = H(z) - I(z, y)
\]
\[
H(z \mid yy') = H(z) - I(z, yy')
\]
where \(I : 2^X \times 2^X \rightarrow [0, \infty)\) is the mutual information. Then we rewrite the inequality as follows
\[
k \geq H(z) - I(z, y) - (H(z) - I(z, yy'))
\]
\[
k \geq H(z) - I(z, y) - H(z) + I(z, yy')
\]
\[
k \geq I(z, yy') - I(z, y),
\]
but mutual information is submodular (see [11] and references therein), and thus \(I(z, yy') \leq I(z, y)\) which means that
\[
I(z, yy') - I(z, y) \leq 0
\]
which concludes the proof since \(k \geq 0\).

\[\Box\]

\section*{B.2 Proof of Theorem 4.5 and Corollaries 4.5.2 and 4.5.1}

\textbf{Proof of Theorem 4.5}

\textit{Proof.} We start by outlining some known relations about MLE and information theory. Consider a MLE of a parameter \(\theta \in \Theta\) with likelihood \(f(\cdot \mid \theta)\) and true value \(\theta^*\); the relation between MLE and information theory can be unravelled by observing that when \(m \rightarrow \infty\)
\[
E_{f(x \mid \theta^*)} \left[ \log f(x \mid \theta^*) - \log f(x \mid \theta) \right] = E_{f(x \mid \theta^*)} \left[ \log \frac{f(x \mid \theta^*)}{f(x \mid \theta)} \right]
\]
\[
= \int f(x \mid \theta^*) \frac{f(x \mid \theta^*)}{f(x \mid \theta)} dx = D_{KL}(f_{\theta^*} \parallel f_{\theta}) \quad (19)
\]
which means that for \(\theta \rightarrow \theta^*\) the Kullback-Leibler divergence \(D_{KL}(f_{\theta^*} \parallel f_{\theta}) \rightarrow a.s. 0\). So, the MLE estimate \(\theta^{MLE} \) moving towards \(\theta^*\) tend to reproduce the same distribution of all the data.

When we compute the MLE on the parameters \(\theta\) and the structure \(A \in \mathcal{A}\) of a model we compute a likelihood \(f(x \mid \theta, A)\) and we assume that the parameters and the structure are independent. This
allows to decouple the problem as two nested MLEs:

for fixed $A$ we maximize $\theta$’s likelihood,

$$\theta_{\text{MLE}} = \arg\max_{\theta \in \Theta} \sum_i \log f(x_i | \theta, A)$$

$$= \arg\min_{\theta \in \Theta} \sum_i -\log f(x_i | \theta, A), \quad (20)$$

and we move towards the MLE structure,

$$A_{\text{MLE}} = \arg\max_{A \in A} \sum_i \log f(x_i | \theta_{A_{\text{MLE}}}^\text{MLE}, A)$$

$$= \arg\min_{A \in A} \sum_i -\log f(x_i | \theta_{A_{\text{MLE}}}^\text{MLE}, A). \quad (21)$$

So we know by eq. (19) that the increments follow the Kullback-Leibler divergence among the discrete distributions

$$\mathbb{E}_{f(x|\theta_{A_{i}}^{\text{MLE}}, A_{i})} \left[ \log f(x | \theta_{A_{i}}^{\text{MLE}}, A_{i}) - \log f(x | \theta_{A_{i}}^{\text{MLE}}, A) \right] = D_{\text{KL}}(f_{A_{i}, \theta_{A_{i}}^{\text{MLE}}} \parallel f_{A_{i}, \theta_{A_{i}}^{\text{MLE}}}) .$$

and that, in the limit of the succession of structures $\{A_{i}\}$ approaching $A$, this quantity almost surely approaches 0.

We can give an interpretation of this in terms of Bayesian updates if we consider $f_{A_{i}, \theta_{A_{i}}^{\text{MLE}}}$ our prior estimate, and $f_{A_{i}, \theta_{A_{i}}^{\text{MLE}}}$ the posterior – both are distributions on all data. We are just saying that the change in the prior to the posterior likelihood is given by the Kullback-Leibler divergence among the distributions, a fact which is known.

We can reduce our theorem to an inequality over the Kullback-Leibler divergence. Recall that we work with the fitness function

$$F_{\Pi, f}(E) = \text{LL}(D | M) - f(M, D) \quad \text{for a BN } M = \langle E, \theta_{\text{MLE}}^{\text{MLE}} \rangle . \quad (22)$$

Now, to make explicit the dependence of $F_{\Pi, f}$ and $\text{LL}$ to the structure of $M$ we adopt the notations $F_{M}$ and $\Delta_{\text{LLM}}$. General submodularity of the fitness function would account in showing that for any $e$ and $A \subseteq B$

$$\Delta_{\text{LLA}(e) | A} - \Delta_{e | A} = \Delta_{e | A} \quad \geq \quad \Delta_{\text{LLB}(e) | B} = \Delta_{\text{LLB}(e) | B} - \Delta_{e | B} .$$

Notice that for a consistent regularizer $f$ we do not need to add the models as subscript to $f$, as the differential of the penalization is a constant

$$\Delta_{e | A} = f(A \cup \{e\}, D) - f(A, D) = f(B \cup \{e\}, D) - f(B, D) = \Delta_{e | B} \geq 0 .$$

So, if we denote by $\mathbb{E}_{f_{A_{i}(e)}}[\Delta_{\text{LLA}(e) | A}]$ the expectation under the distribution associated to structure $A \cup \{e\}$ with the MLE estimates of the parameters, we can just prove

$$D_{\text{KL}}(f_{A_{i}(e)} \parallel f_{A}) = \mathbb{E}_{f_{A_{i}(e)}}[\Delta_{\text{LLA}(e) | A}] \geq \mathbb{E}_{f_{B_{i}(e)}}[\Delta_{\text{LLB}(e) | B}] = D_{\text{KL}}(f_{B_{i}(e)} \parallel f_{B}) .$$
We write the Kullback-Leibler divergence in the context of BNs. Let us elaborate the log-likelihood for a generic model $M$

$$f_M = LL(D \mid M) = \log p(D \mid E, \theta)$$

$$= \sum_y \log p(x = y \mid E, \theta) \quad (y \in D \text{ is a sample, } x = [x_1, \ldots, x_n])$$

$$= \sum_y \sum_{x_i} \log p(x_i = y_i \mid \pi_i = y_{\pi_i}) \quad (y_i \text{ is the } i\text{-th component of } y)$$

where the probability of a single-sample $y$ to be generated by the joint distribution $x$ (we drop $E, \theta$ from the notation) is obtained as follows

$$\log p(x = y) = \log \prod_{x_i} p(x_i = y_i \mid \pi_i = y_{\pi_i}) = \sum_{x_i} \log p(x_i = y_i \mid \pi_i = y_{\pi_i}).$$

Let us introduce the notations $\pi^X_i$ and $p_X(\cdot)$ to denote the parent set and the likelihood function for a model indexed by $X$. The Kullback-Leibler divergence associated to the expectation of the discrete derivative $\Delta_{LL_{A,\{e\}}}$ is hence

$$D_{KL}(f_{A,\{e\}} \parallel f_A) = \sum_y f(y \mid \theta_{MLE_{A,\{e\}}}, A \cup \{e\}) \log \frac{f(y \mid \theta_{MLE_{A,\{e\}}}, A \cup \{e\})}{f(y \mid \theta_{MLE}, A)}$$

$$= \sum_y \sum_{x_i} p_{A,\{e\}}(x_i = y_i \mid \pi^A_{A,\{e\}} = y_{\pi^A_{A,\{e\}}}) \log \sum_{x_i} \frac{p_{A,\{e\}}(x_i = y_i \mid \pi^A_{A,\{e\}} = y_{\pi^A_{A,\{e\}}})}{p_A(x_i = y_i \mid \pi^A = y_{\pi^A})}.$$

We can now decouple the factorized likelihood of a BN. The Kullback-Leibler divergence is additive in the sense that

$$D_{KL}(P \parallel Q) = D_{KL}(P_1 \parallel Q_1) + D_{KL}(P_2 \parallel Q_2)$$

when $P = P_1 P_2$ and $Q = Q_1 Q_2$ are two factorizations of $P$ and $Q$. Consider edge $e : x_i \rightarrow x_j$, and factorize as follows the distribution $f_X$

$$f_X = p_X(x_j = y_j \mid \pi^X_j = y_{\pi^X_j}) \prod_{i \neq j} p_X(x_i = y_i \mid \pi^X_i = y_{\pi^X_i}) = f_i \neq j f_j.$$

which follows since BNs factorize the joint distribution over the variables $x$. Then

$$f_{A,\{e\}} = f_{i \neq j}^{A,\{e\}} f_j^{A,\{e\}}$$

$$f_A = f_{i \neq j}^A f_j^A$$

where we made explicit the terms of the factorization that we multiply (still indexing the structure
$A \cup \{e\}$, and hence

$$D_{KL}(f_{A \cup \{e\}} \| f_A) =$$

$$= D_{KL}(f_{x_j | \pi_j \cup \{x_i\}} \| f_{x_j | \pi_j})$$

$$= \sum_y p_{A \cup \{e\}} (x_j = y_j \mid \pi_j = y_{\pi_j}, x_i = y_i) \log \frac{p_{A \cup \{e\}} (x_j = y_j \mid \pi_j = y_{\pi_j}, x_i = y_i)}{p_A (x_j = y_j \mid \pi_j = y_{\pi_j})}. $$

where the left term is 0 since $A$ and $A \cup \{e\}$ have the same structure for all nodes but $x_j$ – and so the same set of MLE parameters for those nodes. The subscript in $f_{x_j | \pi_j}$ is just to make explicit who the parent set is. Similarly,

$$D_{KL}(f_{B \cup \{e\}} \| f_B) =$$

$$= D_{KL}(f_{x_j | \pi_j \cup \{x_i\}, \pi'_j \cup \{x_i\}} \| f_{x_j | \pi_j \cup \pi'_j}).$$

since $A \subseteq B$ – hence $\pi'_j$ are the parents of $x_j$ in $B$, and not in $A$. Thus, we have reduced our theorem to proving an inequality on the Kullback-Leibler divergence of the parent sets that involve the new edge

$$D_{KL}(f_{A \cup \{e\}} \| f_A) \geq D_{KL}(f_{B \cup \{e\}} \| f_B).$$

Then the proof follows by Lemma B.1 and B.2.

Proof of Corollary 4.5.2.

Proof. The empirical distribution is a consistent estimator for $p$; so for large $m$ $\hat{p}_m \rightarrow_{a.s.} p$ and $E[\hat{p}_m] \rightarrow a.s. E[p]$. Thus, for large enough $m$ we are computing the expectation under a distribution that approaches the true one; by Theorem 4.5 the inequality holds in expectation for $m \rightarrow \infty$.

Proof of Corollary 4.5.1.

Proof. Every submodular function $f$ on $2^M$ is also submodular on $2^{2^M}$.
Agony with no MHC
Agony with Bonferroni
Confidence with FDR
Confidence with Bonferroni
Hill Climbing with
Hill Climbing with

Figure S8: Synthetic tests with different settings from Figure 4: top, $f=AIC$, mid, $\delta = 0.4$, and bottom, continuous variables. In left, for $n$ the number of nodes in the model, we generate $10 \times n$ samples, in right $50 \times n$. 

31
Figure S9: Synthetic tests with binary variables for the BDE Bayesian score. We observe that these simulations, as well as those for other Bayesian scores (Supplementary Figures S10 and S11) show similar trends to the ones discussed in the main text for information-theoretic scoring functions.
Figure S10: Synthetic tests with binary variables for the K2 Bayesian score.
Figure S11: Synthetic tests with Gaussian variables for the BGE Bayesian score.
Figure S12: Violin plots for different p-values $p$ on the model-selection for the alarm network with the agony poset and Bonferroni correction. 100 random datasets are generated with $m = 100$ samples. The same settings of Figure 6 are used.