Entropy-based Pruning for Learning Bayesian Networks using BIC

Cassio P. de Campos[a,b,*], Mauro Scanagatta[c], Giorgio Corani[c], Marco Zaffalon[c]

[a] *Utrecht University, The Netherlands*
[b] *Queen's University Belfast, United Kingdom*
[c] *Istituto Dalle Molle di studi sull'Intelligenza Artificiale (IDSIA), Lugano, Switzerland*

Abstract

For decomposable score-based structure learning of Bayesian networks, existing approaches first compute a collection of candidate parent sets for each variable and then optimize over this collection by choosing one parent set for each variable without creating directed cycles while maximizing the total score. We target the task of constructing the collection of candidate parent sets when the score of choice is the Bayesian Information Criterion (BIC). We provide new non-trivial results that can be used to prune the search space of candidate parent sets of each node. We analyze how these new results relate to previous ideas in the literature both theoretically and empirically. We show in experiments with UCI data sets that gains can be significant. Since the new pruning rules are easy to implement and have low computational costs, they can be promptly integrated into all state-of-the-art methods for structure learning of Bayesian networks.
Keywords: Structure learning; Bayesian networks; BIC; Parent set pruning.

1. Introduction

A Bayesian network [1] is a well-known probabilistic graphical model with applications in a variety of fields. It is composed of (i) an acyclic directed graph (DAG) where each node is associated to a random variable and arcs represent dependencies between the variables entailing the *Markov* condition: every variable is conditionally independent of its non-descendant variables given its parents; and (ii) a set of conditional probability mass functions defined for each variable given its parents in the graph. Their graphical nature makes Bayesian networks excellent models for representing the complex probabilistic relationships existing in many real problems ranging from bioinformatics to law, from image processing to economic risk analysis.

Learning the structure (that is, the graph) of a Bayesian network from complete data is an NP-hard task [2]. We are interested in score-based learning, namely finding the structure which maximizes a score that depends on the data [3]. A typical first step of methods for this purpose is to build a list of suitable candidate parent sets for each one of the n variables of the domain. Later an optimization is run to find one element from each such list in a way that maximizes the total score and does not create directed cycles. This work concerns pruning ideas in order to build those lists. The problem is unlikely to admit a polynomial-time (in n) algorithm, since it is proven to be LOGSNP-hard [4]. Because of that, usually one forces a maximum in-degree (number of parents per node) k and then simply computes the score of all parent sets that contain up to k parents. A worth-mention exception is the greedy search of the K2 algorithm [5].

A high in-degree implies a large search space for the optimization and thus increases the possibility of finding better structures. On the other hand, it requires higher computational time, since there are $\Theta(n^k)$ candidate parent sets for a bound of k if an exhaustive search is performed. Our contribution is to provide new rules for pruning sub-optimal parent sets when dealing with the *Bayesian Information Criterion* score [6], one of the most used score functions in the literature. We

*Corresponding author

devise new theoretical bounds that can be used in conjunction with currently published ones [7]. The new results provide tighter bounds on the maximum number of parents of each variable in the optimal graph, as well as new pruning techniques that can be used to skip large portions of the search space without any loss of optimality. Moreover, the bounds can be efficiently computed and are easy to implement, so they can be promptly integrated into existing software for learning Bayesian networks and imply immediate computational gains.

The paper is divided as follows. Section 2 presents the problem, some background and notation. Section 3 describes the existing results in the literature, and Section 4 contains the theoretical developments for the new bounds and pruning rules. Section 5 shows empirical results comparing the new results against previous ones, and finally some conclusions are given in Section 6.

2. Structure learning of Bayesian networks

Consider the problem of learning the structure of a Bayesian Network from a complete data set of $N \geq 2$ instances $\mathcal{D} = \{D_1, ..., D_N\}$. The set of $n \geq 2$ categorical random variables is denoted by $\mathcal{X} = \{X_1, ..., X_n\}$ (each variable has at least two categories). The state space of X_i is denoted Ω_{X_i} and a joint space for $\mathcal{X}_1 \subseteq \mathcal{X}$ is denoted by $\Omega_{\mathcal{X}_1} = \times_{X \in \mathcal{X}_1} \Omega_X$ (and with a slight abuse $|\Omega_\varnothing| = 1$ containing a null element). The goal is to find the best DAG $\mathcal{G} = (V, E)$, where V is the collection of nodes (associated one-to-one with the variables in $\mathcal{X}$) and E is the collection of arcs. E can be represented by the (possibly empty) set of parents $\Pi_1, ..., \Pi_n$ of each node/variable.

Different score functions can be used to assess the quality of a DAG. This paper regards the *Bayesian Information Criterion* (or simply BIC) [6], which asymptotically approximates the posterior probability of the DAG. The BIC score is *decomposable*, that is, it can be written as a sum of the scores of each variable and its parent set:

$$\mathrm{BIC}(\mathcal{G}) = \sum_{i=1}^{n} \mathrm{BIC}(X_i|\Pi_i) = \sum_{i=1}^{n} \left(\mathrm{LL}(X_i|\Pi_i) + \mathrm{Pen}(X_i|\Pi_i)\right) ,$$

where $\mathrm{LL}(X_i|\Pi_i)$ denotes the log-likelihood of X_i and its parent set:

$$\mathrm{LL}(X_i|\Pi_i) = \sum_{\pi \in \Omega_{\Pi_i}} \sum_{x \in \Omega_{X_i}} N_{x,\pi} \log_b \hat{\theta}_{x|\pi} ,$$

where the base $b \geq 2$ is usually taken as natural or 2. We will make it clear when the result depends on such base. Moreover, $\hat{\theta}_{x|\pi}$ is the maximum likelihood estimate of the conditional probability $P(X_i = x|\Pi_i = \pi)$, that is, $N_{x,\pi}/N_\pi$; N_π and $N_{x,\pi}$ represents, respectively, the number of times $(\Pi_i = \pi)$ and $(X_i = x \wedge \Pi_i = \pi)$ appear in the data set (if π is null, then $N_\pi = N$ and $N_{x,\pi} = N_x$). In the case with no parents, we use the notation $\mathrm{LL}(X_i) = \mathrm{LL}(X_i|\varnothing)$. $\mathrm{Pen}(X_i|\Pi_i)$ is the complexity penalization for X_i and its parent set:

$$\mathrm{Pen}(X_i|\Pi_i) = -\frac{\log_b N}{2}(|\Omega_{X_i}| - 1)|\Omega_{\Pi_i}| ,$$

again with the notation $\mathrm{Pen}(X_i) = \mathrm{Pen}(X_i|\varnothing)$.

For completeness, we present the definition of (conditional) mutual information. Let $\mathcal{X}_1$, $\mathcal{X}_2$, $\mathcal{X}_3$ be pairwise disjoint subsets of $\mathcal{X}$. Then

$$\mathrm{I}(\mathcal{X}_1, \mathcal{X}_2|\mathcal{X}_3) = \mathrm{H}(\mathcal{X}_1|\mathcal{X}_3) - \mathrm{H}(\mathcal{X}_1|\mathcal{X}_2 \cup \mathcal{X}_3)$$

(unconditional version is obtained with $\mathcal{X}_3 = \varnothing$), and (the sample estimate of) entropy is defined as usual: $\mathrm{H}(\mathcal{X}_1|\mathcal{X}_2) = \mathrm{H}(\mathcal{X}_1 \cup \mathcal{X}_2) - \mathrm{H}(\mathcal{X}_2)$ and

$$\mathrm{H}(\mathcal{X}_1) = -\sum_{x \in \Omega_{\mathcal{X}_1}} \frac{N_x}{N} \log_b \left(\frac{N_x}{N}\right) .$$

(x runs over the configurations of $\mathcal{X}_1$.) Since $\hat{\theta}_x = N_x/N$, it is clear that $N \cdot \mathrm{H}(\mathcal{X}_1|\mathcal{X}_2) = -\mathrm{LL}(\mathcal{X}_1|X_2)$ for any disjoint subsets $\mathcal{X}_1, \mathcal{X}_2 \subseteq \mathcal{X}$.

The ultimate goal is to find $\mathcal{G}^* \in \operatorname{argmax}_{\mathcal{G}} \mathrm{BIC}(\mathcal{G})$ (we avoid equality because there might be multiple optima). We assume that if two DAGs $\mathcal{G}_1$ and $\mathcal{G}_2$ have the same score, then we prefer the graph with fewer arcs. The usual first step to achieve such goal is the task of finding the *candidate parent sets* for a given variable X_i (obviously a candidate parent set cannot contain X_i itself). This task regards constructing the list L_i of parent sets Π_i for X_i alongside their scores $\mathrm{BIC}(X_i|\Pi_i)$. Without any restriction, there are 2^{n-1} possible parent sets, since every subset of $\mathcal{X} \setminus \{X_i\}$ is a candidate. Each score computation costs $\Theta(N \cdot (1 + |\Pi_i|))$, and the number of score computations becomes quickly prohibitive with the increase of n. In order to avoid losing global optimality, we must guarantee that L_i contains candidate parent sets that cover those in an optimal DAG. For instance, if we apply a bound k on the number of parents that a variable can have, then the size of

$$L_i = \{\langle \Pi_i, \mathrm{BIC}(X_i|\Pi_i)\rangle \mid |\Pi_i| \leq k\}$$

is $\Theta(n^k)$, but we might lose global optimality (this is the case if any optimal DAG would have more than k parents for X_i). Irrespective of that, this pruning is not enough if n is large. Bounds greater than 2 can already become prohibitive. For instance, a bound of $k = 2$ is adopted in [8] when dealing with its largest data set (diabetes), which contains 413 variables. One way of circumventing the problem is to apply pruning rules which allow us to discard/ignore elements of L_i in such a way that an optimal parent set is never discarded/ignored.

3. Pruning rules

The simplest pruning rule one finds in the literature states that if a candidate subset has better score than a candidate set, then such candidate set can be safely ignored, since the candidate subset will never yield directed cycles if the candidate set itself does not yield cycles [9, 10]. By safely ignoring/discarding a candidate set we mean that we are still able to find an optimal DAG (so no accuracy is lost) even if such parent set is never used. This is formalized as follows.

Lemma 1. *(Theorem 1 in [7], but also found elsewhere [9].) Let Π^* be a candidate parent set for the node $X \in \mathcal{X}$. Suppose there exists a parent set Π such that $\Pi \subset \Pi^*$ and $\mathrm{BIC}(X|\Pi) \geq \mathrm{BIC}(X|\Pi^*)$. Then Π^* can be safely discarded from the list of candidate parent sets of X.*

This result can be also written in terms of the list of candidate parent sets. In order to find an optimal DAG for the structure learning problem, it is sufficient to work with

$$L_i = \{\langle \Pi_i, \mathrm{BIC}(X_i|\Pi_i)\rangle \mid \forall \Pi_i' \subset \Pi_i : \mathrm{BIC}(X_i|\Pi_i) > \mathrm{BIC}(X_i|\Pi_i')\}.$$

Unfortunately there is no way of applying Lemma 1 without computing the scores of all candidate sets, and hence it provides no speed up for building the list (it is nevertheless useful for later optimizations, but that is not the focus of this work).

There are however pruning rules that can reduce the computation time for finding L_i and that are still safe.

Lemma 2. *Let $\Pi \subset \Pi'$ be candidate parent sets for $X \in \mathcal{X}$. Then $\mathrm{LL}(X|\Pi) \leq \mathrm{LL}(X|\Pi')$, $\mathrm{H}(X|\Pi) \geq \mathrm{H}(X|\Pi')$ and $\mathrm{Pen}(X|\Pi) > \mathrm{Pen}(X|\Pi')$.*

Proof. The inequalities follow directly from the definitions of log-likelihood, entropy and penalization. □

Lemma 3. *(Theorem 4 in [7].[1]) Let $X \in \mathcal{X}$ be a node with $\Pi \subset \Pi^*$ two candidate parent sets, such that $\mathrm{BIC}(X|\Pi) \geq \mathrm{Pen}(X|\Pi^*)$. Then Π^* and all its supersets can be safely ignored when building the list of candidate parent sets for X.*

[1] There is an imprecision in the Theorem 4 of [7], since t_i as defined there does not account for the constant of BIC/AIC while in fact it should. In spite of that, their desired result is clear. We present a proof for completeness.

Proof. Let $\Pi' \supseteq \Pi^*$. By Lemma 2, we have $\text{Pen}(X|\Pi^*) \geq \text{Pen}(X|\Pi')$ (equality only if $\Pi^* = \Pi'$). Then $\text{BIC}(X|\Pi) \geq \text{Pen}(X|\Pi^*) \Rightarrow \text{BIC}(X|\Pi) \geq \text{Pen}(X|\Pi') \Rightarrow \text{BIC}(X|\Pi) - \text{BIC}(X|\Pi') \geq -\text{LL}(X|\Pi')$, and we have $-\text{LL}(X|\Pi') \geq 0$, so Lemma 1 suffices to conclude the proof. □

Note that $\text{BIC}(X|\Pi) \geq \text{Pen}(X|\Pi^*)$ can as well be written as $\text{LL}(X|\Pi) \geq \text{Pen}(X|\Pi^*) - \text{Pen}(X|\Pi)$, and if $\Pi^* = \Pi \cup \{Y\}$ for some $Y \notin \Pi$, then it can be written also as $\text{LL}(X|\Pi) \geq (|\Omega_Y| - 1)\text{Pen}(X|\Pi)$. The reasoning behind Lemma 3 is that the maximum improvement that we can have in BIC score by inserting new parents into Π would be achieved if $\text{LL}(X|\Pi)$, which is a non-positive value, grew all the way to zero, since the penalization only gets worse with more parents. If $\text{LL}(X|\Pi)$ is already close enough to zero, then the loss in the penalty part cannot be compensated by the gain of likelihood. The result holds for every superset because both likelihood and penalty are monotone with respect to increasing the number of parents.

4. Novel pruning rules

In this section we devise novel pruning rules by exploiting the empirical entropy of variables. We later demonstrate that such rules are useful to ignore elements while building the list L_i that cannot be ignored by Lemma 3, hence tightening the pruning results available in the literature. In order to achieve our main theorem, we need some intermediate results.

Lemma 4. *Let $\Pi = \Pi' \cup \{Y\}$ for $Y \notin \Pi'$, with Π, Π' candidate parent sets for $X \in \mathcal{X}$. Then* $\text{LL}(X|\Pi) - \text{LL}(X|\Pi') \leq N \cdot \min\{\text{H}(X|\Pi'); \text{H}(Y|\Pi')\}$.

Proof. This comes from simple manipulations and known bounds to the value of conditional mutual information.

$$\begin{aligned}
\text{LL}(X|\Pi) - \text{LL}(X|\Pi') &= N \cdot (\text{H}(X|\Pi') - \text{H}(X|\Pi)) \leq N \cdot \text{H}(X|\Pi')\,. \\
\text{LL}(X|\Pi) - \text{LL}(X|\Pi') &= N \cdot \text{I}(X, Y|\Pi') \\
&= N \cdot (\text{H}(Y|\Pi') - \text{H}(Y|\Pi' \cup \{X\})) \leq N \cdot \text{H}(Y|\Pi')\,.
\end{aligned}$$

□

Theorem 1. *Let $X \in \mathcal{X}$, and Π^* be a parent set for X. Let $Y \in \mathcal{X} \setminus \Pi^*$ such that $N \cdot \min\{\text{H}(X|\Pi^*); \text{H}(Y|\Pi^*)\} \leq (1 - |\Omega_Y|)\text{Pen}(X|\Pi^*)$. Then the parent set $\Pi = \Pi^* \cup \{Y\}$ and all its supersets can be safely ignored when building the list of candidate parents sets for X.*

Proof. We have that

$$\begin{aligned}
\text{BIC}(X|\Pi) &= \text{LL}(X|\Pi) + \text{Pen}(X|\Pi) \\
&\leq \text{LL}(X|\Pi^*) + N \cdot \min\{\text{H}(X|\Pi^*); \text{H}(Y|\Pi^*)\} + \text{Pen}(X|\Pi) \\
&\leq \text{LL}(X|\Pi^*) + (1 - |\Omega_Y|)\text{Pen}(X|\Pi^*) + \text{Pen}(X|\Pi) \\
&= \text{LL}(X|\Pi^*) + \text{Pen}(X|\Pi^*) - \text{Pen}(X|\Pi) + \text{Pen}(X|\Pi) = \text{BIC}(X|\Pi^*)\,.
\end{aligned}$$

First step is the definition of BIC, second step uses Lemma 4 and third step uses the assumption of this theorem. Therefore, Π can be safely ignored (Lemma 1). Now take any $\Pi' \supset \Pi$. Let $\Pi'' = \Pi' \setminus \{Y\}$. It is immediate that $N \cdot \min\{\text{H}(X|\Pi^*); \text{H}(Y|\Pi^*)\} \leq (1 - |\Omega_Y|)\text{Pen}(X|\Pi^*) \Rightarrow N \cdot \min\{\text{H}(X|\Pi''); \text{H}(Y|\Pi'')\} \leq (1 - |\Omega_Y|)\text{Pen}(X|\Pi'')$, since $\Pi^* \subset \Pi''$ and hence $-\text{Pen}(X|\Pi'') > -\text{Pen}(X|\Pi^*)$. The theorem follows by the same arguments as before, applied to Π' and Π''. □

The rationale behind Theorem 1 is that if the data do not have entropy in amount enough to beat the penalty function, then there is no reason to continue expanding the parent set candidates. Theorem 1 can be used for pruning the search space of candidate parent sets without having to compute their BIC scores. However, we must have available the conditional entropies $\text{H}(X|\Pi^*)$ and $\text{H}(Y|\Pi^*)$. The former is usually available, since $-N \cdot \text{H}(X|\Pi^*) = \text{LL}(X|\Pi^*)$, which is used to compute $\text{BIC}(X|\Pi^*)$ (and it is natural to assume that such score has been already computed at

the moment Theorem 1 is checked). Actually, this bound amounts exactly to the previous result in the literature (see for example [7]):

$$\begin{aligned}
&N \cdot \mathrm{H}(X|\Pi^*) \leq (1 - |\Omega_Y|)\mathrm{Pen}(X|\Pi^*) \iff \\
&\mathrm{LL}(X|\Pi^*) \geq \mathrm{Pen}(X|\Pi^* \cup \{Y\}) - \mathrm{Pen}(X|\Pi^*) \iff \\
&\mathrm{BIC}(X|\Pi^*) \geq \mathrm{Pen}(X|\Pi^* \cup \{Y\})\,.
\end{aligned}$$

By Theorem 1 we know that $\Pi^* \cup \{Y\}$ and any superset can be safely ignored, which is the very same condition as in Lemma 3. The novelty in Theorem 1 comes from the term $\mathrm{H}(Y|\Pi^*)$. If such term is already computed (or if it needs to be computed irrespective of this bound computation, and thus we do not lose time computing it for this purpose only), then we get (almost) for free a new manner to prune parent sets. In case this computation of $\mathrm{H}(Y|\Pi^*)$ is not considered worthwhile, or if we simply want a faster approach to prune parent sets, we can resort to a more general version of Theorem 1, as given by Theorem 2.

Theorem 2. *Let $X \in \mathcal{X}$, and Π^*, Π' be parent sets for X with $\Pi' \subseteq \Pi^*$. Let $Y \in \mathcal{X} \setminus \Pi^*$ such that $N \cdot \min\{\mathrm{H}(X|\Pi'); \mathrm{H}(Y|\Pi')\} \leq (1 - |\Omega_Y|)\mathrm{Pen}(X|\Pi^*)$. Then the parent set $\Pi = \Pi^* \cup \{Y\}$ and all its supersets can be safely ignored when building the list of candidate parents sets for X.*

Proof. It is well-known (see Lemma 2) that $\mathrm{H}(X|\Pi^*) \leq \mathrm{H}(X|\Pi')$ and $\mathrm{H}(Y|\Pi^*) \leq \mathrm{H}(Y|\Pi')$ for any X,Y,$\Pi' \subseteq \Pi^*$ as defined in this theorem, so the result follows from Theorem 1. □

An important property of Theorem 2 when compared to Theorem 1 is that all entropy values regard subsets of the current parent set at our own choice. For instance, we can choose $\Pi' = \varnothing$ and so they become entropies of single variables, which can be precomputed efficiently in total time $O(N \cdot n)$. Another option at this point, if we do not want to compute $\mathrm{H}(Y|\Pi^*)$ and assuming the cache of Y has been already created, would be to quickly inspect the cache of Y to find the most suitable subset of Π^* to plug into Theorem 2. Moreover, with Theorem 2, we can prune the search space of a variable X without evaluating the likelihood of parent sets for X (just by using the entropies), and so it could be used to guide the search even before any heavy computation is done. The main novelty in Theorems 1 and 2 is to make use of the (conditional) entropy of Y.

This new pruning approach is not trivially achievable by previous existing bounds for BIC. It is worth noting the relation with previous work. The restriction of Theorem 2 can be rewritten as:

$$\begin{aligned}
&N \cdot \min\{\mathrm{H}(X|\Pi'); \mathrm{H}(Y|\Pi')\} \leq (1 - |\Omega_Y|)\mathrm{Pen}(X|\Pi^*) \iff \\
&N \cdot \min\{\mathrm{H}(X|\Pi'); \mathrm{H}(Y|\Pi')\} + \mathrm{LL}(X|\Pi^*) \leq -\mathrm{Pen}(X|\Pi^* \cup \{Y\}) + \mathrm{BIC}(X|\Pi^*).
\end{aligned}$$

Note that the condition for Lemma 3 (known from literature) is exactly $-\mathrm{Pen}(X|\Pi^* \cup \{Y\}) + \mathrm{BIC}(X|\Pi^*) \geq 0$. Hence, Theorem 2 will be effective (while the previous rule in Lemma 3 will not) when $-\mathrm{Pen}(X|\Pi^* \cup \{Y\}) + \mathrm{BIC}(X|\Pi^*) < 0$, and so when $N \cdot \min\{\mathrm{H}(X|\Pi'); \mathrm{H}(Y|\Pi')\} + \mathrm{LL}(X|\Pi^*) < 0$. Intuitively, the new bound of Theorem 2 might be more useful when the parent set being evaluated is poor (hence $\mathrm{LL}(X|\Pi^*)$ is low) while the result in Lemma 3 plays an important role when the parent set being evaluated is good (and so $\mathrm{LL}(X|\Pi^*)$ is high).

The result of Theorem 2 can also be used to bound the maximum number of parents in any given candidate parent set. While the asymptotic result is already implied by previous work [7, 12], we obtain the finer and interesting result of Theorem 3.

Theorem 3. *There is an optimal structure such that variable $X \in \mathcal{X}$ has at most*

$$\max_{Y \in \mathcal{X} \setminus \{X\}} \left\lceil 1 + \log_2 \left(\frac{\min\{\mathrm{H}(X); \mathrm{H}(Y)\}}{(|\Omega_X| - 1)(|\Omega_Y| - 1)} \right) + \log_2 N - \log_2 \log_b N \right\rceil^+$$

parents, where $\lceil \cdot \rceil^+$ denotes the smallest natural number greater than or equal to its argument.

Proof. If $\Pi = \varnothing$ is the optimal parent for X, then the result trivially follows since $|\Pi| = 0$. Now take Π such that $Y \in \Pi$ and $\Pi^* = \Pi \setminus \{Y\}$. Since $|\Pi| = |\Pi^*| + 1$ and $|\Omega_{\Pi^*}| \geq 2^{|\Pi^*|}$, we have $|\Pi| \leq \log_2 |\Omega_{\Pi^*}| + 1$. Now, if

$$\begin{aligned}
&\log_2 |\Omega_{\Pi^*}| \geq 1 + \log_2 \left(\frac{\min\{\mathrm{H}(X); \mathrm{H}(Y)\}}{(|\Omega_X| - 1)(|\Omega_Y| - 1)} \right) + \log_2 N - \log_2 \log_b N \iff \\
&\log_2 |\Omega_{\Pi^*}| \geq \log_2 \left(\frac{2 \min\{\mathrm{H}(X); \mathrm{H}(Y)\}}{(|\Omega_X| - 1)(|\Omega_Y| - 1)} \cdot \frac{N}{\log_b N} \right) \iff \\
&N \cdot \min\{\mathrm{H}(X); \mathrm{H}(Y)\} \leq \frac{\log_b N}{2} \cdot |\Omega_{\Pi^*}|(|\Omega_X| - 1)(|\Omega_Y| - 1) \iff \\
&N \cdot \min\{\mathrm{H}(X); \mathrm{H}(Y)\} \leq (1 - |\Omega_Y|) \cdot \mathrm{Pen}(X|\Pi^*) \,,
\end{aligned}$$

then by Theorem 2 (used with $\Pi' = \varnothing$) every superset of Π^* containing Y can be safely ignored, and so it would be Π. Therefore,

$$|\Pi| \leq 1 + \log_2 |\Omega_{\Pi^*}| < 1 + 1 + \log_2 \left(\frac{\min\{\mathrm{H}(X); \mathrm{H}(Y)\}}{(|\Omega_X| - 1)(|\Omega_Y| - 1)} \right) + \log_2 N - \log_2 \log_b N \,,$$

and since $|\Pi|$ is a natural number, the result follows by applying the same reasoning for every $Y \in \mathcal{X} \setminus \{X\}$. □

Corollary 1 is demonstrated for completeness, since it is implied by previous work (see for instance [7]; a similar result is implied in [12], but the last passage is flawed there and the bound seems slightly better). It is nevertheless presented here in more detailed terms and without an asymptotic function.

Corollary 1. *There is an optimal structure such that each variable has at most* $\lceil 1 + \log_2 N - \log_2 \log_b N \rceil$ *parents.*

Proof. By Theorem 3, we have that Π can be a parent of a node X only if

$$\begin{aligned}
|\Pi| &\leq \max_{Y \in \mathcal{X} \setminus \{X\}} \left\lceil 1 + \log_2 \left(\frac{\min\{\mathrm{H}(X); \mathrm{H}(Y)\}}{(|\Omega_X| - 1)(|\Omega_Y| - 1)} \right) + \log_2 N - \log_2 \log_b N \right\rceil^+ \\
&\leq \max_{Y \in \mathcal{X} \setminus \{X\}} \left\lceil 1 + \log_2 \left(\frac{\mathrm{H}(Y)}{(|\Omega_X| - 1)(|\Omega_Y| - 1)} \right) + \log_2 N - \log_2 \log_b N \right\rceil^+ \\
&\leq \max_{Y \in \mathcal{X} \setminus \{X\}} \max \left\lceil 1 + \log_2 \left(\frac{\log_b |\Omega_Y|}{(|\Omega_X| - 1)(|\Omega_Y| - 1)} \right) + \log_2 N - \log_2 \log_b N \right\rceil^+ \\
&\leq \left\lceil 1 + \log_2 \left(\frac{1}{(|\Omega_X| - 1)} \right) + \log_2 N - \log_2 \log_b N \right\rceil^+ \\
&\leq \lceil 1 + \log_2 N - \log_2 \log_b N \rceil^+ \\
&\leq \lceil 1 + \log_2 N - \log_2 \log_b N \rceil \,,
\end{aligned}$$

since it is assumed that $N \geq 2$ and $b \geq 2$. □

Theorem 3 can be used to bound the number of parent sets per variable, even before computing parent sets for them, with the low computation cost of computing the empirical entropy of each variable once (hence overall cost of $O(n \cdot N)$ time). We point out that Theorem 3 can provide effective bounds (considerably smaller than $\lceil 1 + \log_2 N - \log_2 \log_b N \rceil$) on the number of parents for specific variables, particularly when number of states is high and entropies are low, as we will see in the next section.

5. Experiments

We run experiments using a collection of data sets from the UCI repository [11]. Table 1 shows the data set names, number of variables n and number of data points N. In the same table, we show the maximum number of parents that a node can have, according to the new result of Theorem 3, as well as the old result from the literature (which we present in Corollary 1). The old bound is global, so a single number is given in column 5, while the new result of Theorem 3 implies a different maximum number of parents per node. We use the notation *bound (number of times)*, with the bound followed by the number of nodes for which the new bound reached that value, in parenthesis (so all numbers in parenthesis in a row should sum to n of that row). We see that the gains with the new bounds are quite significant and can prune great parts of the search space further than previous results.

			Bound on number of parents	
Dataset	n	N	Theorem 3	Corollary 1
glass	8	214	6 (7), 3 (1)	6
diabetes	9	768	7 (9)	8
tic-tac-toe	10	958	6 (10)	8
cmc	10	1473	8 (3), 7 (7)	9
breast-cancer	10	286	6 (4), 5 (2), 4 (1), 3 (3)	7
solar-flare	12	1066	7 (4), 6 (1), 5 (5), 3 (1), 2 (1)	8
heart-h	12	294	6 (6), 5 (3), 4 (2), 3 (1)	7
vowel	14	990	8 (12), 4 (2)	8
zoo	17	101	5 (10), 4 (6), 2 (1)	5
vote	17	435	7 (15), 6 (2)	7
segment	17	2310	9 (16), 6 (1)	9
lymph	18	148	5 (8), 4 (8), 3 (2)	6
primary-tumor	18	339	6 (9), 5 (7), 4 (1), 2 (1)	7
vehicle	19	846	7 (18), 6 (1)	8
hepatitis	20	155	5 (18), 4 (2)	6
colic	23	368	6 (8), 5 (12), 4 (3)	7
autos	26	205	6 (16), 5 (3), 4 (1), 3 (5), 1 (1)	6
flags	29	194	6 (5), 5 (7), 4 (7), 3 (7), 2 (3)	6

Table 1: Maximum number of parents that nodes have using new (column 4) and previous bounds (column 5). In column 4, we list the bound on number of parents followed by how many nodes have that bound in parenthesis (the new theoretical results obtain a specific bound per node, while previous results obtain a single global bound).

Our second set of experiments compares the activation of Theorems 1, 2, and 3 in pruning the search space for the construction of the list of candidate parent sets. Tables 2 to 4 (in the end of this document) present the results as follows. Columns one to four contain, respectively, the data set name, number of variables, number of data points and maximum in-degree (in-d) that we impose (a maximum in-degree is imposed so as we can compare the obtained results among different approaches). The fifth column, named $|S|$, presents the total number of parent sets that need to be evaluated by the brute-force procedure (taking into consideration the imposed maximum in-degree). Column 6 has the average time to run the algorithms (there is actually no significant difference between the algorithms' times in our experiments). Columns 7 to 13 present the number of times that different pruning results are activated when exploring the whole search space. Larger numbers means that more parent sets are ignored (even without being evaluated). The naming convention for the pruning algorithms as used on those columns is:

*Alg*1 Application of Theorem 1 using $\mathrm{H}(X|\Pi^*)$ in the expression of the rule (instead of the minimization), where X is the variable for which we are building the list and Π is the current parent set being explored. This is equivalent to the previous rule in the literature, as presented in this paper in Lemma 3.

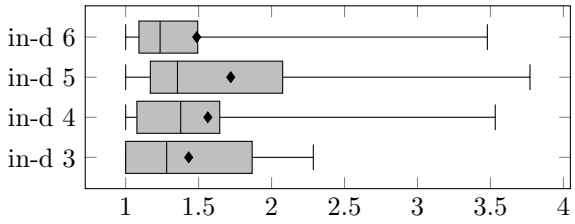

Figure 1: Ratio between pruned candidates using *Alg*1+*Alg*2 (which theoretically subsumes *Alg*3 and *Alg*4) divided by pruned candidates using prune approach *Alg*1 alone (since it was the previous literature result), for different values of maximum in-degree. Greater than one means better than *Alg*1. Results over 18 data sets. Averages are marked with a diamond. Whiskers are at min./max. and box has the 2nd and 3rd quartiles.

*Alg*2 Application of Theorem 1 using $\mathrm{H}(Y|\Pi^*)$ in the expression of the rule (instead of the minimization), where X is the variable for which we are building the list and Y is the variable just to be inserted in the parent set Π^* that is being explored. This is the new pruning rule which makes most use of entropy, but it may be slower than the others (since conditional entropies might need to be evaluated, if they were not yet).

*Alg*3 Application of Theorem 2 using $H(X)$ in the formula, that is, with $\Pi' = \varnothing$ (and instead of the minimization). This is a slight improvement to the known rule in the literature regarding the maximum number of parents of a variable and is very fast, since it does not depend on evaluating any parent sets.

*Alg*4 Application of Theorem 2 using $H(Y)$ in the formula, that is, with $\Pi' = \varnothing$ (and instead of the minimization). This is a different improvement to the known rule in the literature regarding the maximum number of parents of a variable and is very fast, since it does not depend on evaluating any parent sets.

We also present the combined number of pruning obtained by some of these ideas when they are applied together. Of particular interest is column 8 with *Alg*1+*Alg*2, as it shows the largest amount of pruning that is possible, albeit more computationally costly because of the (possibly required) computations for *Alg*2 (even though we have observed no significant computational time difference within our experiments). This is also presented graphically in the boxplot of Figure 1, where the values for the 18 data sets are summarized and the amount of pruning is divided by the pruning of *Alg*1, and so a ratio above one shows (proportional) gain with respect to the previous literature pruning rule.

Column 13 of Tables 2 to 4 have the pruning results (number of ignored candidates) for *Alg*1 and *Alg*4 together, since this represents the pruning obtained by the old rule plus the new rule given by Theorem 2 in such a way that absolutely no extra computational cost takes place (and moreover it subsumes approach *Alg*3, since *Alg*1 is theoretically superior to *Alg*3). Again, this is summarized in the boxplot of Figure 2 over the 18 data sets and the values are divided by the amount of pruning of *Alg*1 alone, so values above one show the (proportional) gain with respect to the previous literature rule.

As we can see in more detail in Tables 2 to 4, the gains with the new pruning ideas are significant in many circumstances. Moreover, there is no extra computational cost for applying *Alg*3 and *Alg*4, so one should always apply those rules while deciding selectively whether to employ prune *Alg*2 or not (we recall that one can tune that rule by exploiting the flexibility of Theorem 2 and searching for a subset that is already available in the computed lists, so a more sophisticated pruning scheme is also possible – we experiment here with a simple idea that does not bring extra computational time, and leave for future work the design of other strategies).

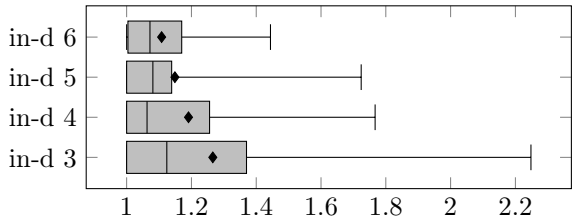

Figure 2: Ratio between pruned candidates using $Alg1{+}Alg4$ (which theoretically subsumes $Alg3$ and $Alg4$) divided by pruned candidates using prune approach $Alg1$ alone, for different values of maximum in-degree. Greater than one means better than $Alg1$. Results over 18 data sets. Averages are marked with a diamond. Whiskers are at min./max. and box has the 2nd and 3rd quartiles.

6. Conclusions

This paper presents new non-trivial pruning rules to be used with the Bayesian Information Criterion (BIC) score for learning the structure of Bayesian networks. The derived theoretical bounds extend previous results in the literature and can be promptly integrated into existing solvers with minimal effort and computational costs. They imply faster computations without losing optimality. The very computationally efficient version of the new rules imply gains of around 20% with respect to previous work, according to our experiments, while the most computationally demanding pruning achieves around 50% more pruning than before. Pruning rules for other widely used scores such as the Bayesian Dirichlet equivalent uniform (BDeu) have been devised [13] and some researchers conjecture that they cannot be improved. Similarly, we conjecture that further bounds for the BIC score are unlikely to exist unless for some particular cases and situations. This can be studied in a future work, as well as means to devise smart strategies to tune the theorem parameters and improve their pruning capabilities.

Acknowledgments

Work partially supported by the Swiss NSF grant n. 200021_146606 /1 and ns. IZKSZ2_162188.

References

[1] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann Publishers Inc., 1988.

[2] D. M. Chickering, D. Heckerman, C. Meek, Large-sample learning of Bayesian networks is NP-hard, Journal of Machine Learning Resesearch 5 (2014) 1287–1330.

[3] D. Heckerman, D. Geiger, D. Chickering, Learning Bayesian networks: The combination of knowledge and statistical data, Machine Learning 20 (1995) 197–243.

[4] M. Koivisto, Parent assignment is hard for the MDL, AIC, and NML costs, in: Proceedings of the 19th Annual Conference on Learning Theory, Springer-Verlag, 2006, pp. 289–303.

[5] G. F. Cooper, E. Herskovits, A Bayesian method for the induction of probabilistic networks from data, Machine Learning 9 (4) (1992) 309–347.

[6] G. Schwarz, Estimating the dimension of a model, The Annals of Statistics 6 (1978) 461–464.

[7] C. P. de Campos, Q. Ji, Efficient structure learning of Bayesian networks using constraints, Journal of Machine Learning Research 12 (2011) 663–689.

[8] M. Bartlett, J. Cussens, Integer linear programming for the Bayesian network structure learning problem, Artificial Intelligence 24 (2017) 258–271.

[9] M. Teyssier, D. Koller, Ordering-based search: A simple and effective algorithm for learning Bayesian networks, in: Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence, 2005, pp. 584–590.

[10] C. P. de Campos, Z. Zeng, Q. Ji, Structure learning of Bayesian networks using constraints, in: Proceedings of the 26th Annual International Conference on Machine Learning, ACM, 2009, pp. 113–120.

[11] M. Lichman, UCI machine learning repository (2013).
URL `http://archive.ics.uci.edu/ml`

[12] J. Tian, A Branch-and-Bound Algorithm for MDL Learning Bayesian Networks, in Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, 2000, pp. 580–588.

[13] C. P. de Campos, Q. Ji, Properties of Bayesian Dirichlet score to learn Bayesian network structures, in: Proceedings of the 24th AAAI Conference on Artificial Intelligence, AAAI Press, 2010, pp. 431–436.

Dataset	n	N	in-d	$\|S\|$	time(sec)	*Alg*1	*Alg*2	*Alg*1 + *Alg*2	*Alg*3	*Alg*4	*Alg*3 + *Alg*4	*Alg*1 + *Alg*4
			3	504	1	0	0	0	0	0	0	0
glass	8	214	4	784	1	114	66	154	0	0	0	114
			5	952	1	240	192	280	105	126	126	240
			6	1008	1	294	248	336	154	175	175	294
			7	1016	2	302	256	344	162	183	183	302
			3	828	1	0	0	0	0	0	0	0
diabetes	9	768	4	1458	2	0	0	0	0	0	0	0
			5	1962	4	0	0	0	0	0	0	0
			6	2214	6	0	0	0	0	0	0	0
			7	2286	8	0	0	0	0	0	0	0
			3	1290	1	624	624	704	611	581	671	684
breast-cancer	10	286	4	2550	4	1785	1804	1902	1765	1733	1855	1874
			5	3810	14	3041	3061	3161	3017	2987	3109	3130
			6	4650	30	3881	3901	4001	3857	3827	3949	3970
			7	5010	48	4241	4261	4361	4217	4187	4309	4330
			3	1290	2	0	0	0	0	0	0	0
cmc	10	1473	4	2550	9	30	81	106	7	27	34	53
			5	3810	31	419	768	913	291	561	696	743
			6	4650	53	1152	1592	1753	1002	1363	1520	1567
			7	5010	91	1512	1952	2113	1362	1723	1880	1927
			3	1290	1	0	0	0	0	0	0	0
tic-tac-toe	10	958	4	2550	5	0	0	0	0	0	0	0
			5	3810	11	659	1114	1244	504	504	504	659
			6	4650	23	1499	1954	2084	1344	1344	1344	1499
			7	5010	39	1859	2314	2444	1704	1704	1704	1859
			3	2772	1	206	374	481	196	348	471	473
heart-h	12	294	4	6732	5	1873	3016	3277	1696	2774	3156	3179
			5	12276	16	6473	8404	8707	6250	8054	8506	8535
			6	17820	45	11984	13947	14251	11709	13597	14050	14079
			7	21780	123	15944	17907	18211	15669	17557	18010	18039

Table 2: Pruning results for multiple UCI data sets. Columns contain, respectively: data set name, number of variables, number of data points, maximum imposed in-degree, size of search space, average time to run an algorithm (no significant different between them), followed by the number of pruned parent sets when considering (a combination of) different pruning rules (see the list of pruning rules for more details).

Dataset	n	N	in-d	$\|S\|$	time(sec)	$Alg1$	$Alg2$	$Alg1$ + $Alg2$	$Alg3$	$Alg4$	$Alg3$ + $Alg4$	$Alg1$ + $Alg4$
			3	2772	4	872	1476	1740	739	1295	1570	1626
solar-flare	12	1066	4	6732	28	3817	5280	5634	3442	4929	5344	5426
			5	12276	157	9150	10821	11178	8732	10461	10885	10967
			6	17820	538	14692	16365	16722	14268	16005	16429	16511
			7	21780	1389	18652	20325	20682	18228	19965	20389	20471
			3	2772	1	206	374	481	196	348	471	473
heart-h	12	294	4	6732	5	1873	3016	3277	1696	2774	3156	3179
			5	12276	16	6473	8404	8707	6250	8054	8506	8535
			6	17820	45	11984	13947	14251	11709	13597	14050	14079
			7	21780	123	15944	17907	18211	15669	17557	18010	18039
			3	5278	4	288	500	614	132	132	264	400
vowel	14	990	4	15288	28	1718	2500	2854	1232	1232	1364	1830
			5	33306	149	10257	13202	14247	8162	4994	9086	11161
			6	57330	534	27608	31682	32727	24794	17930	27566	29641
			7	81354	1491	47672	51746	52791	44066	37994	47630	49705
			3	11832	3	0	0	0	0	0	0	0
vote	17	435	4	42772	17	0	0	0	0	0	0	0
			5	117028	110	577	852	1429	0	0	0	577
			6	253164	667	14939	39946	51968	0	0	0	14939
			7	447644	3039	163532	234426	246448	11440	80080	91520	183677
			3	11832	17	0	0	0	0	0	0	0
segment	17	2310	4	42772	112	201	309	506	0	0	0	201
			5	117028	670	3983	7628	10674	0	0	0	3983
			6	253164	2976	32679	69813	79254	0	0	0	32679
			7	447644	11381	133330	226619	245782	0	0	0	133330
			3	11832	2	1717	2396	2660	735	686	1337	2229
zoo	17	101	4	42772	12	14353	17076	20060	8015	9058	10437	14871
			5	117028	58	76677	91222	94311	37226	50281	54663	81892
			6	253164	215	212813	227358	230447	173362	186417	190799	218028
			7	447644	539	407293	421838	424927	367842	380897	385279	412508
			3	14994	3	3320	4680	5918	2892	3552	4874	5222
lymph	18	148	4	57834	28	34206	41374	45262	29982	35202	40208	42205
			5	169218	187	140227	152420	156538	133806	144338	150656	153059
			6	391986	1074	362942	375188	379306	355314	367101	373424	375827
			7	742050	5016	713006	725252	729370	705378	717165	723488	725891

Table 3: Pruning results for multiple UCI data sets. Columns contain, respectively: data set name, number of variables, number of data points, maximum imposed in-degree, size of search space, average time to run an algorithm (no significant different between them), followed by the number of pruned parent sets when considering (a combination of) different pruning rules (see the list of pruning rules for more details).

Dataset	n	N	in-d	$\|S\|$	time(sec)	*Alg*1	*Alg*2	*Alg*1 + *Alg*2	*Alg*3	*Alg*4	*Alg*3 + *Alg*4	*Alg*1 + *Alg*4
			3	14994	5	2202	2648	3097	2177	2293	3034	3049
primary-tumor	18	339	4	57834	65	14518	19410	21237	14207	18682	20552	20827
			5	169218	425	69480	102175	110881	66935	89698	100427	101894
			6	391986	2419	262650	322756	333064	241655	306499	320211	322604
			7	742050	9748	610976	672820	683128	581423	656563	670275	672668
			3	18753	8	3	3	6	0	0	0	3
vehicle	19	846	4	76893	65	697	1111	1764	0	0	0	697
			5	239685	422	10863	23352	32440	0	0	0	10863
			6	592401	3330	104254	215785	258003	0	0	0	104254
			7	1197057	10803	474794	787023	846604	222768	31824	254592	501193
			3	23180	3	0	0	0	0	0	0	0
hepatitis	20	155	4	100700	26	0	0	0	0	0	0	0
			5	333260	183	21692	64380	81809	0	0	0	21692
			6	875900	1024	444629	606978	624448	217056	529704	537096	562889
			7	1883660	3208	1452389	1614738	1632208	1224816	1537464	1544856	1570649
			3	41239	21	3108	4622	5896	2362	3515	4536	5067
colic	23	368	4	209484	273	96090	110871	124974	85987	94637	111045	116851
			5	815166	2512	657339	691757	715181	629257	653090	687159	698902
			6	2531265	28377	2367440	2407325	2431074	2337376	2365549	2401648	2413789
			3	68250	20	20751	23699	27655	18178	18413	23162	25555
autos	26	205	4	397150	229	246272	269886	292669	192752	213929	245722	280668
			5	1778530	3792	1450595	1560143	1607068	1333004	1395969	1469070	1544272
			6	6383130	42555	5964376	6164743	6211668	5622016	5839681	5937534	6113490
			3	106720	46	62817	69264	75214	60196	62868	70472	71892
flags	29	194	4	700495	1240	593809	623134	636559	577776	596922	619080	624253
			5	3550586	23605	3393251	3458658	3475307	3338989	3408696	3434987	3445679

Table 4: Pruning results for multiple UCI data sets. Columns contain, respectively: data set name, number of variables, number of data points, maximum imposed in-degree, size of search space, average time to run an algorithm (no significant different between them), followed by the number of pruned parent sets when considering (a combination of) different pruning rules (see the list of pruning rules for more details).