On Pruning for Score-Based Bayesian Network Structure Learning

On Pruning for Score-Based Bayesian Network Structure Learning

Alvaro H. C. Correia *
Utrecht University
Utrecht, The Netherlands
a.h.chaimcorreia@uu.nl

James Cussens *
University of York
York, United Kingdom
james.cussens@york.ac.uk

Cassio de Campos *
Utrecht University
Utrecht, The Netherlands
c.decampos@uu.nl

Abstract

Many algorithms for score-based Bayesian network structure learning (BNSL) take as input a collection of potentially optimal parent sets for each variable in a data set. Constructing these collections naively is computationally intensive since the number of parent sets grows exponentially with the number of variables. Therefore, pruning techniques are not only desirable but essential. While effective pruning exists for the Bayesian Information Criterion (BIC), current results for the Bayesian Dirichlet equivalent uniform (BDeu) score reduce the search space very modestly, hampering the use of (the often preferred) BDeu. We derive new non-trivial theoretical upper bounds for the BDeu score that considerably improve on the state of the art. Since the new bounds are efficient and easy to implement, they can be promptly integrated into many BNSL methods. We show that gains can be significant in multiple UCI data sets so as to highlight practical implications of the theoretical advances.

1 Introduction

A Bayesian network [19] is a widely used probabilistic graphical model. It is composed of (i) a *structure* defined by a directed acyclic graph (DAG) where each node is associated with a random variable, and where 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 collection of conditional probability distributions defined for each variable given its parents in the graph. Their graphical nature make Bayesian networks ideal for complex probabilistic relationships existing in many real-world problems [8].

Bayesian network structure learning (BNSL) with complete data is NP-hard [3]. We tackle score-based learning, that is, finding the structure maximising a given (data-dependent) score [14]. In particular, we focus on the *Bayesian Dirichlet equivalent uniform* (BDeu) score [4], which corresponds to the log probability of the structure given (multinomial) data and a uniform prior on structures: The BDeu score is decomposable, that is, it can be written as a sum of *local scores* of the domain variables: $\mathrm{BDeu}(\mathcal{G}) = \sum_{i \in V} \mathrm{LBDeu}(i, S_i)$, where LBDeu is the local score function, $V = \{1, \ldots, n\}$ is the set of (indices of) variables in the dataset, which is in correspondence with nodes of the Bayesian network to be learned, and $S_i \subseteq V^{\setminus i}$, with $V^{\setminus i} = V \setminus \{i\}$, is the parent set of node i in the DAG structure $\mathcal{G}$. A common approach divides the problem into two steps:

1. CANDIDATE PARENT SET IDENTIFICATION: For each variable of the domain, find a suitable collection of candidate parent sets and their local scores.
2. STRUCTURE OPTIMISATION: Given the collection of candidate parent sets, choose a parent set for each variable so as to maximise the overall score while avoiding directed cycles.

Preprint. Under review.
*All authors contributed equally to this work.

This paper concerns pruning ideas to help solve candidate parent set identification. The problem is unlikely to admit a polynomial-time (in n) algorithm (it is proven to be LOGSNP-Hard [16] for BIC), so usually one chooses a maximum in-degree d (number of parents per node) and then computes the score of parent sets with in-degree at most d. Increasing the maximum in-degree can considerably improve the chances of finding better structures but requires higher computational time, since there are $\Theta(n^d)$ candidate parent sets (per variable) for a bound of d if an exhaustive search is performed, and 2^{n-1} without an in-degree constraint. For instance, $d > 2$ can already become prohibitive [1]. Our contribution is to provide new theoretical upper bounds for the local scores in order to prune non-optimal parent sets without ever having to compute their scores. Such upper bounds can then be used together with any searching approach [2, 5, 10, 12, 15, 17, 21, 22]. These bounds are efficiently computed and easy to implement, so they can be easily integrated into existing software for BNSL.

While the main goal of this paper is to provide new theoretical upper bounds that are provably superior to the state of the art [6, 9, 10], we also investigate how such bounds are effective in practice. This is done by performing experiments with multiple datasets from the *UCI Machine Learning Repository* [13]. The results support our motivation for new tighter bounds, in particular, by allowing us to learn more efficiently without a maximum in-degree d, which may be especially important in domains with complex relations.

The paper is organised as follows. Section 2 provides the notation and required definitions, as well as a brief description of the current best bound for BDeu in the literature. Section 4 presents an improved bound whose derivation follows the same approach as the existing one, but exploits properties of the score function to get tighter results. This new bound is effective in many datasets, as we show in the experiments. Still, it does not capture all cases and other bounds can be devised. Section 5 looks at the problem from a new angle and introduces an upper bound based on a (tweaked) maximum likelihood estimation. These bounds are finally combined and empirically compared against each other in Section 6. Section 7 concludes the paper and gives directions for future research. The proofs of intermediate lemmas and corollaries are left to the appendix for brevity.

2 Definitions and notation

First of all, because the collection of scores are computed independently for each variable in the dataset (BDeu is decomposable), we drop i from the notation and use simply $\mathrm{LBDeu}(S)$ to refer to the score of node i with parent set S. We need some further notation:

- $c(i)$ is the state space of variable i, and $c(S)$ is the set of all joint instantiations/configurations of the random variables in $S \subseteq V$, that is, $c(S) = \times_{j \in S} c(j)$ the Cartesian product of the state space of involved variables. Moreover, $q(S) = |c(S)|$, and we abuse notation to say $q(i) = |c(i)|$.
- The data $\mathcal{D}$ is a *multiset* (that is, repetitions are allowed) of elements from $c(V)$, with $\mathcal{D}^S$ the reduction of dimension of $\mathcal{D}$ only to the part regarding variables in $S \subseteq V$ (note that $\mathcal{D} = \mathcal{D}^V$), and $\mathcal{D}^S(j_{S'}) \subseteq \mathcal{D}^S$, with $j_{S'} \in c(S')$, are the elements of $\mathcal{D}^S$ such that $\mathcal{D}^{S \cap S'} = j_{S'}^{S \cap S'}$. The subscript of $j_{S'}$ is omitted if clear from the context. We use the notation $\mathcal{D}_u$ instead of $\mathcal{D}$ to denote the set of unique elements from the corresponding multiset $\mathcal{D}$.
- For $j \in c(S)$, we define $n_j = |\mathcal{D}^S(j)|$, that is, the number of occurrences of j in $\mathcal{D}^S$.
- $\vec{\alpha}_j = (\alpha_{j,k})_{k \in c(i)}$ is the prior vector for parent set $S \subseteq V^{\setminus i}$ under configuration $j \in c(S)$, which in the BDeu score satisfies $\alpha_{j,k} = \alpha_{\text{ess}}/q(S \cup \{i\})$, with α_{ess} as the equivalent sample size, a user parameter to define the strength of the prior.

Let $\Gamma_\alpha(x) = \frac{\Gamma(x+\alpha)}{\Gamma(\alpha)}$ for x nonnegative integer and $\alpha > 0$ (Γ denotes the Gamma function). Denote $\sum_{k \in c(i)} \alpha_{j,k} = \alpha_{\text{ess}}/q(S)$ by α_j. The local score for i with parent set $S \subseteq V^{\setminus i}$ can be written as

$$\mathrm{LBDeu}(S) = \sum_{j \in c(S)} \mathrm{LLBDeu}(S, j), \text{ and } \mathrm{LLBDeu}(S, j) = -\log \Gamma_{\alpha_j}(n_j) + \sum_{k \in c(i)} \log \Gamma_{\alpha_{j,k}}(n_{j,k}) .$$

That is, $\mathrm{LBDeu}(S)$ is a sum of $q(S)$ values each of which is specific to a particular instantiation of the variables S. We call such values *local local BDeu scores (llB)*. In particular, $\mathrm{LLBDeu}(S, j) = 0$ if its $n_j = 0$, so we can concentrate only on those which actually appear in the data:

$$\mathrm{LBDeu}(S) = \sum_{j \in \mathcal{D}_u^S} \mathrm{LLBDeu}(S, j) .$$

3 Pruning in Candidate Parent Set Identification

The pruning of parent sets rests on the (simple) observation that a parent set cannot be optimal if one of its subsets has a higher score [20]. Thus, when learning Bayesian networks from data using BDeu, it is important to have an upper bound $\mathrm{ub}(S) \geqslant \max_{T:T\supset S} \mathrm{LBDeu}(T)$ so as to potentially prune a whole area of the search space at once. Ideally, one would like an upper bound that is tight and cheap to compute, so that one can score parent sets S incrementally, and at the same time check whether it is worth 'expanding' S: if $\mathrm{ub}(S)$ is not greater than $\max_{R:R\subseteq S} \mathrm{LBDeu}(R)$, then it is unnecessary to expand S. In practice, however, there is a trade-off between these two desiderata. With that in mind, we can define candidate parent set identification more formally:

> CANDIDATE PARENT SET IDENTIFICATION: For each variable $i \in V$, find a collection of parent sets $\mathcal{L}_i$, such that $\mathcal{L}_i = \{S \subseteq V^{\setminus i} : S' \subset S \Rightarrow \mathrm{LBDeu}(S') < \mathrm{LBDeu}(S)\}$.

Unfortunately, we cannot predict the elements of $\mathcal{L}_i$ and have to compute the scores for a list $L_i \supseteq \mathcal{L}_i$. The practical benefit of our bounds is to reduce $|L_i|$, and consequently to lower the computational cost, while ensuring that $L_i \supseteq \mathcal{L}_i$. Before presenting the best known upper bound [6, 9, 10], we present a lemma on the variation of counts with expansions of the parent set.

Lemma 1. *For $S \subseteq T \subseteq V^{\setminus i}$, $j_S \in \mathcal{D}_u^S$ and $j_T \in \mathcal{D}_u^T$ with $j_T^S = j_S$, $|\mathcal{D}_u^{T\cup\{i\}}| \geqslant |\mathcal{D}_u^{S\cup\{i\}}|$ and $|\mathcal{D}_u^{T\cup\{i\}}(j_T)| \leqslant |\mathcal{D}_u^{S\cup\{i\}}(j_S)|$.*

As an example, consider the small dataset of Table 1. The number of non-zero counts never decreases as we add a new variable to the parent set of variable $i = 3$. With $S = \{1\}$ and $T = \{1, 2\}$, we have $|\mathcal{D}_u^{S\cup\{i\}}| = 3$ and $|\mathcal{D}_u^{T\cup\{i\}}| = 4$. Conversely, the number of (unique) occurrences compatible with a given instantiation of the parent set never increases with its expansion: for example with $j_S = (\mathrm{var}_1 : 1)$ and $j_T = (\mathrm{var}_1 : 1, \mathrm{var}_2 : 1)$, we have $|\mathcal{D}_u^{S\cup\{i\}}(j_S)| = 2$ and $|\mathcal{D}_u^{T\cup\{i\}}(j_T)| = 2$.

Table 1: Example of data $\mathcal{D}$, its reductions by parent sets $S = \{1\}$ and $T = \{1, 2\}$, and the number of unique occurrences compatible with $j_S \in \mathcal{D}_u^S$ and $j_T, j'_T \in \mathcal{D}_u^T$, with $j_T^S = {j'}_T^S = j_S$. The child variable is $i = 3$, and we have $j_S = (\mathrm{var}_1 : 1)$, $j_T = (\mathrm{var}_1 : 1, \mathrm{var}_2 : 1)$, $j'_T = (\mathrm{var}_1 : 1, \mathrm{var}_2 : 0)$.

$\mathcal{D}$			$\mathcal{D}_u^{S\cup\{i\}}$		$\mathcal{D}_u^{T\cup\{i\}}$			$\mathcal{D}_u^{S\cup\{i\}}(j_S)$		$\mathcal{D}_u^{T\cup\{i\}}(j_T)$			$\mathcal{D}_u^{T\cup\{i\}}(j'_T)$		
1	2	3	1	3	1	2	3	1	3	1	2	3	1	2	3
0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0
1	0	0	1	0	1	0	0	1	1	1	1	1			
1	1	0	1	1	1	1	0								
1	1	1			1	1	1								

Theorem 1 (Bound f [7, 10])**.** *Let $S \subseteq V^{\setminus i}$, $j \in \mathcal{D}_u^S$, and let $f(S, j) = -|\mathcal{D}_u^{S\cup\{i\}}(j)| \log q(i)$. Then, $\mathrm{LLBDeu}(S, j) \leqslant f(S, j)$. Moreover, if $\mathrm{LBDeu}(S') \geqslant \sum_{j\in\mathcal{D}_u^S} f(S, j) = f(S)$ for some $S' \subset S$, then all $T \supseteq S$ are not in $\mathcal{L}_i$.*

This means we compute the number of non-zero counts per instantiation, $|\mathcal{D}_u^{S\cup\{i\}}(j)|$, and we 'gain' $\log q(i)$ for each of them. Note that $f(S) = -|\mathcal{D}_u^{S\cup\{i\}}| \log q(i)$, which by Lemma 1 is monotonically non-increasing over expansions of the parent set S. Hence $f(S)$ is not only an upper bound on $\mathrm{LBDeu}(S)$ but also on $\mathrm{LBDeu}(T)$ for every $T \supseteq S$. Bound f is cheap to compute but is unfortunately too loose. We derive much tighter upper bounds which are actually bounds on these llBs. Thus, an upper bound for a local BDeu score is obtained by simple addition as just described. We will derive an upper bound on $\mathrm{LLBDeu}(S, j)$ (where $n_j > 0$) by considering instantiation counts for the *full* parent set $V^{\setminus i}$, the parent set which includes all possible parents for child i. We call these *full instantiation counts*. Evidently, the number of full parent instantiations $q(V^{\setminus i})$ grows exponentially with $|V|$, but it is linear in $|\mathcal{D}|$ when we consider only the unique elements $\mathcal{D}_u^{V^{\setminus i}}$.

4 Exploiting Gamma function properties

First, we extend the current state-of-the-art upper bound of Theorem 1 by exploiting some properties of the Gamma function. For that, we need some intermediate results, where we assume $\alpha > 0$.

Lemma 2. *Let x be a positive integer. Then $\Gamma_\alpha(0) = 1$ and $\log \Gamma_\alpha(x) = \sum_{\ell=0}^{x-1} \log(\ell + \alpha)$.*

Lemma 3. *For x positive integer and $v \geqslant 1$, it holds that $\log\big(\Gamma_\alpha(x)/\Gamma_{\alpha/v}(x)\big) \geqslant \log v$.*

Lemma 4. *Let x, y be non-negative integers with $x + y > 0$.*

$$\begin{cases} \Gamma_\alpha(x+y) = \Gamma_\alpha(x)\,\Gamma_\alpha(y) & \text{if } x \cdot y = 0\,, \\ \Gamma_\alpha(x+y) \geqslant \Gamma_\alpha(x)\,\Gamma_\alpha(y)\,(1 + y/\alpha) & \text{otherwise.} \end{cases}$$

Corollary 1. *Let $x_1, \ldots, x_k$ be a list of non-negative integers in decreasing order with $x_1 > 0$, then*

$$\Gamma_\alpha\left(\sum_{l=1}^{k} x_l\right) \geqslant \prod_{l=1}^{k} \Gamma_\alpha(x_l) \prod_{l=1}^{k'-1} (1 + x_l/\alpha)\,,$$

where $k' \leqslant k$ is the last positive integer in the list (the second product disappears if $k' = 1$).

Lemma 5. *For $S \subseteq V^{\setminus i}$ and $j \in \mathcal{D}_u^S$, assume that $\vec{n}_j = (n_{j,k})_{k \in c(i)}$ are in decreasing order over $k = 1, \ldots, q(i)$ (this is without loss of generality, since we can name and process them in any order). Then for any $\alpha \geqslant \alpha_j = \alpha_{ess}/q(S)$, we have*

$$\mathrm{LLBDeu}(S, j) \leqslant f(S, j) + g(S, j, \alpha), \text{ with } g(S, j, \alpha) = -\sum_{l=1}^{k'-1} \log(1 + n_{j,l}/\alpha)\,,$$

and $k' \leqslant k$ is the largest index such that $n_{j,k'} > 0$.

The difference here is the summation from the gap of the super-multiplicativity of Γ (Lemma 4 and Corollary 1). That extra term gives us a tighter bound on $\mathrm{LLBDeu}(S, j)$, but $g(S) = f(S) + \sum_{j \in \mathcal{D}_u^S} g(S, j, \alpha)$ is no longer monotonic over expansions of S (though monotone in α). Hence, $g(S)$ is not an upper bound on $\mathrm{LBDeu}(T)$ for every $T \supseteq S$, and we need further results on $g(S, j, \alpha)$.

Lemma 6. *For $S \subseteq T \subseteq V^{\setminus i}$, $j_T \in \mathcal{D}_u^T$, and $j_S \in \mathcal{D}_u^S$ with $j_T^S = j_S$, $f(T, j_T) \geqslant f(S, j_S)$ and $g(T, j_T, \alpha) \geqslant g(S, j_S, \alpha)$.*

Theorem 2 (Bound $\underline{g}$)**.** *Let $S \subseteq V^{\setminus i}$, $j_S \in \mathcal{D}_u^S$, $\underline{g}(S, j_S) = \min_{j \in \mathcal{D}_u^{V^{\setminus i}}:\, j^S = j_S} g(V^{\setminus i}, j, \alpha_{ess}/q(S))$. Then $\mathrm{LLBDeu}(S, j_S) \leqslant f(S, j_S) + \underline{g}(S, j_S)$. Also, if $\mathrm{LBDeu}(S') \geqslant (f(S) + \sum_{j_S \in \mathcal{D}_u^S} \underline{g}(S, j_S)) = \underline{g}(S)$ for some $S' \subset S$, then all $T \supseteq S$ are not in $\mathcal{L}_i$.*

Proof. First we prove that $f(S, j_S) + \underline{g}(S, j_S)$ is an upper bound for $\mathrm{LLBDeu}(S, j_S)$. From Lemma 6, if we take any instantiation of the fully expanded parent set, $j \in \mathcal{D}_u^{V^{\setminus i}} : j^S = j_S$, we have that $g(S, j_S, \alpha) \leqslant g(V^{\setminus i}, j, \alpha)$ for any α. As Lemma 6 is valid for every full instantiation j, we can take the minimum over them to get the tightest bound. From Lemma 5, $\mathrm{LLBDeu}(S, j_S) \leqslant f(S, j_S) + \underline{g}(S, j_S)$. Now, if we sum all the llBs, we obtain the second part of the theorem for S. Finally, we need to show that this second part holds for any $T \supset S$, which follows from $f(T) \leqslant f(S)$ (as the total number of non-zero counts only increases, by Lemma 1) and

$$\sum_{j_T \in \mathcal{D}_u^T} \underline{g}(T, j_T) = \sum_{j_S \in \mathcal{D}_u^S} \left(\sum_{j_T \in \mathcal{D}_u^T:\, j_T^S = j_S} \underline{g}(T, j_T) \right) \leqslant \sum_{j_S \in \mathcal{D}_u^S} \underline{g}(S, j_S)\,.$$

That holds as $\underline{g}(T, j_T) \leqslant 0$ and, with $j_T^S = j_S$, at least one term $\underline{g}(T, j_T)$ is smaller than $\underline{g}(S, j_S)$, as their minimisation spans the same full instantiations (and $g(\cdot, \cdot, \alpha)$ is non-decreasing on α). □

5 Exploiting the likelihood function

Bound $\underline{g}$ (Theorem 2) was based on the best full instantiation $j \in \mathcal{D}_u^{V^{\setminus i}}$ that is compatible with an llB of the parent set S. Knowing function g is monotonic over parent set sizes, we could look at an instantiation of the fully extended parent set to derive a bound for the llB of S and all its supersets. Even though the results are valid for every full instantiation, we can only compute Bound $\underline{g}$ using one of them at a time. The new bound of this section comes from the realisation that it is possible to exploit all full instantiations to derive a valid bound on the llB of S. For that purpose, we need some properties of inferences with the Dirichlet-multinomial distribution and conjugacy.

The BDeu score is simply the log marginal probability of the observed data given suitably chosen Dirichlet priors over the parameters of a BN structure. Consequently, llBs are intimately connected to the Dirichlet-multinomial conjugacy. Given a Dirichlet prior $\vec{\alpha}_j = (\alpha_{j,1}, \ldots, \alpha_{j,q(i)})$, the probability of observing data $\mathcal{D}_{\vec{n}_j}$ with counts $\vec{n}_j = (n_{j,1}, \ldots, n_{j,q(i)})$ is:

$$\log \Pr(\mathcal{D}_{\vec{n}_j}|\vec{\alpha}_j) = \log \int_p \Pr(\mathcal{D}_{\vec{n}_j}|p)\Pr(p|\vec{\alpha}_j)dp\,,$$

where the first distribution under the integral is multinomial and the second is Dirichlet. Note that

$$\log \int_p \Pr(\mathcal{D}_{\vec{n}_j}|p)\Pr(p|\vec{\alpha}_j)dp \leqslant \max_p \log \Pr(\mathcal{D}_{\vec{n}_j}|p), \tag{1}$$

since $\int_p \Pr(p|\vec{\alpha}_j)dp = 1$. Note also that llBs are not the probability of observing sufficient statistics counts, but of a particular dataset, that is, there is no multinomial coefficient which would consider all the permutations yielding the same sufficient statistics. Therefore, we may devise a bound based on the maximum (log-)likelihood estimation.

Lemma 7. *Let $S \subseteq V^{\setminus i}$ and $j \in \mathcal{D}_u^S$. Then* $\mathrm{LLBDeu}(S, j) \leqslant ML(\vec{n}_j)$*, where we have that* $ML(\vec{n}_j) = \sum_{k\in c(i)} n_{j,k} \log(n_{j,k}/n_j)$*. ($0 \log 0 = 0$.)*

Corollary 2. *Let $S \subseteq V^{\setminus i}$ and $j_S \in \mathcal{D}_u^S$. Then* $\mathrm{LLBDeu}(S, j_S) \leqslant \sum_{j\in\mathcal{D}_u^{V^{\setminus i}}:\, j^S = j_S} ML(\vec{n}_j)$*.*

We can improve further on this bound of Corollary 2 by considering llBs as a function h of α for fixed $\vec{n}_j$, since we can study and exploit the shape of their curves.

$$h_{\vec{n}_j}(\alpha) = -\log \Gamma_\alpha(n_j) + \sum_{k\in c(i)} \log \Gamma_{\alpha/q(i)}(n_{j,k})\,.$$

Lemma 8. *If $\nexists k : n_{j,k} = n_j$, then $h_{\vec{n}_j}$ is a concave function for positive $\alpha \leqslant 1$.*

The concavity of $h_{\vec{n}_j}$ is useful for the following reason.

Lemma 9. *Let $S \subseteq V^{\setminus i}$ and $j \in \mathcal{D}_u^{V^{\setminus i}}$ such that $\nexists k : n_{j,k} = n_j$. If $\alpha \leqslant q(S)$ and $\frac{\partial h_{\vec{n}_j}}{\partial \alpha}(\alpha/q(S))$ is non-negative then $h_{\vec{n}_j}(\alpha/q(T)) \leqslant h_{\vec{n}_j}(\alpha/q(S))$ for every $T \supseteq S$.*

The final step to improve the bound is to consider any score for a parent set as a function of the (log-)probabilities over full mass functions.

Theorem 3. *Let $S \subseteq V^{\setminus i}$ and $j_S \in \mathcal{D}_u^S$. Then* $\mathrm{LLBDeu}(S, j_S) \leqslant \log \Pr(\mathcal{D}_{\vec{n}_{j^\star}}|\vec{\alpha}_{j_S}) + \sum_{j\in\mathcal{D}_u^{V^{\setminus i}}:\, j\neq j^\star} ML(\vec{n}_j)$*, where* $j^\star = \arg\min_{j\in\mathcal{D}_u^{V^{\setminus i}}} \log \Pr(\mathcal{D}_{\vec{n}_j}|\vec{\alpha}_{j_S})$*.*

Proof. We rewrite $n_{j_S,k}$ as the sum of counts from full mass functions: $n_{j_S,k} = \sum_{j\in\mathcal{D}_u^{V^{\setminus i}}:\, j^S = j_S} n_{j,k}$. Thus, $\mathrm{LLBDeu}(S, j_S)$ is the log probability $\log \Pr(\mathcal{D}_{\vec{n}_{j_S}}|\vec{\alpha}_{j_S})$ of observing a data sequence with counts $\vec{n}_{j_S} = (\sum_{j\in\mathcal{D}_u^{V^{\setminus i}}:\, j^S = j_S} n_{j,k})_{k\in c(i)}$ under the Dirichlet-multinomial with parameter vector $\vec{\alpha}_{j_S}$. Assume an arbitrary order for the full mass functions related to elements in $\{j \in \mathcal{D}_u^{V^{\setminus i}} : j^S = j_S\}$ and name them $j_1, \ldots, j_w$, with $w = |\{j \in \mathcal{D}_u^{V^{\setminus i}} : j^S = j_S\}|$. Exploiting the conjugacy multinomial-Dirichlet we can express this probability as a product of conditional probabilities:

$$\Pr(\mathcal{D}_{\vec{n}_{j_S}}|\vec{\alpha}_{j_S}) = \prod_{\ell=1}^{w} \Pr\left(\mathcal{D}_{\vec{n}_{j_\ell}} \middle| \sum_{t=1}^{\ell-1} \vec{n}_{j_t} + \vec{\alpha}_{j_S}\right),$$

$$\mathrm{LLBDeu}(S, j_S) = \sum_{\ell=1}^{w} \log \Pr\left(\mathcal{D}_{\vec{n}_{j_\ell}} \middle| \sum_{t=1}^{\ell-1} \vec{n}_{j_t} + \vec{\alpha}_{j_S}\right) \leqslant \log \Pr(\vec{n}_{j_1}|\vec{\alpha}_{j_S}) + \sum_{t=2}^{w} \mathrm{ML}(\vec{n}_{j_t}).$$

These are obtained by applying Expression (1) to all but the first term. Since the choice of the order is arbitrary, we can do it in our best interest and the theorem is obtained. □

While the bound of Theorem 3 is valid for S, it gives no assurances about its supersets T, so it is of little direct use (if we need to compute it for every $T \supset S$, then it is better to compute the scores themselves). To address that we replace the first term of the right-hand side summation with a proper upper bound.

Theorem 4 (Bound $\underline{h}$). *Let $S \subseteq V^{\setminus i}$, $\alpha = \alpha_{ess}/q(S)$, $j_S \in \mathcal{D}_u^S$, and $\overline{h}_{\vec{n}_j}(\alpha) = h_{\vec{n}_j}(\alpha)$ if $\alpha \leqslant 1$ and $\frac{\partial h_{\vec{n}_j}}{\partial \alpha}(\alpha) \geqslant 0$, and zero otherwise. Let*

$$\underline{h}(S, j_S) = \min_{\substack{j \in \mathcal{D}_u^{V^{\setminus i}}: \\ j^S = j_S}} \Big(-\mathit{ML}(\vec{n}_j) + \min\{\mathit{ML}(\vec{n}_j);\ f(V^{\setminus i}, j) + g(V^{\setminus i}, j, \alpha);\ \overline{h}_{\vec{n}_j}(\alpha)\} \Big) + \sum_{\substack{j \in \mathcal{D}_u^{V^{\setminus i}}: \\ j^S = j_S}} \mathit{ML}(\vec{n}_j). \tag{2}$$

Then $\mathrm{LLBDeu}(S, j_S) \leqslant \underline{h}(S, j_S)$. *Moreover, if* $\mathrm{LBDeu}(S') \geqslant \sum_{j_S \in \mathcal{D}_u^S} \underline{h}(S, j_S) = \underline{h}(S)$ *for some $S' \subset S$, then S and all its supersets are not in $\mathcal{L}_i$.*

Proof. For the parent set S, the bound based on $\mathrm{ML}(\vec{n}_j)$ (that is, using the first option in the inner minimisation) is valid by Corollary 2. The other two options make use of Theorem 3 and their own results: the bound on $f(V^{\setminus i}, j) + g(V^{\setminus i}, j, \alpha)$ is valid by Lemma 6, while the bound based on $\overline{h}_{\vec{n}_j}(\alpha)$ comes from Lemma 9, and thus the result holds for S. Take $T \supset S$. It is straightforward that

$$\mathrm{LBDeu}(T) \leqslant \sum_{j_T \in \mathcal{D}_u^T} \underline{h}(T, j_T) = \sum_{j_S \in \mathcal{D}_u^S} \left(\sum_{j_T \in \mathcal{D}_u^T:\ j_T^S = j_S} \underline{h}(T, j_T) \right) \leqslant \sum_{j_S \in \mathcal{D}_u^S} \underline{h}(S, j_S),$$

since $\sum_{j_T \in \mathcal{D}_u^T:\ j_T^S = j_S} \underline{h}(T, j_T) \leqslant \underline{h}(S, j_S)$, because both sides run over the same full instantiations and the right-hand side use the tighter minimisation of Expression (2) only once, while the left-hand side can use that tighter minimisation once every j_T, and Lemmas 6 and 9 ensure that the computed values $f(V^{\setminus i}, j) + g(V^{\setminus i}, j, \alpha)$ and $\overline{h}_{\vec{n}_j}(\alpha)$ are valid for T. □

We point out that the mathematical results may seem harder to use in practice than they actually are. Computing $\underline{g}(S)$ and $\underline{h}(S)$ to prune a parent set S and all its supersets can be done in linear time, since one pass through the data is enough to collect and process all required counts (AD-trees [18] can be used to get even greater speedups). Since the computation of a score already takes linear time in the number of data samples, we have a cheap bounds which are provably superior to the current state-of-the-art pruning for BDeu. Finally, we also point out that bounds $\underline{g}$ and $\underline{h}$ prune the search spaces differently, as their independent theoretical derivations suggest. Therefore, we combine both to get a tighter bound which we call $\underline{\mathrm{C4}} = \min\{\underline{g}\,;\, \underline{h}\}$. Their differences are illustrated in the sequel.

6 Experiments

To analyse the empirical gains of the new bounds, we computed the list of candidate parent sets for each variable in multiple UCI datasets [13]. In all experiments, we set $\alpha_{\text{ess}} = 1$ and discretise all continuous variables by their median value. To provide an idea of the processing time, small datasets ($n \leqslant 10$) took less than few minutes to complete, while larger ones ($n \geqslant 20$) took around one day per variable (if using a single modern core). The main method is presented in Algorithm 1. Parent sets are explored in order of size (outermost loop), and for each (non-pruned) parent set S, we verify if it has no subset which is better than itself before including it in the resulting set, and then we expand it by adding an extra parent, so long as the pruning criterion is not met. This algorithm is presented in simplified terms: it is possible to cache most of the results to speed up computations.

Algorithm 1 Parent Set Identification

Input: $(i, V, \mathcal{D}, \text{in-d})$. **Output:** $\mathcal{L}_i$.
$\mathcal{L}_i \leftarrow \{\varnothing\},\ L_i \leftarrow \{\varnothing\},\ d \leftarrow 0$.
$b(\mathcal{L}_i, T) = \max_{S \in \mathcal{L}_i:\ S \subset T} \mathrm{LBDeu}(S)$.
while $d \leqslant$ in-d **do**
  for $S \in L_i :\ |S| = d$ **do**
    $\mathcal{L}_i \leftarrow \mathcal{L}_i \cup \{S\}$ **if** $\mathrm{LBDeu}(S) > b(\mathcal{L}_i, S)$.
    $L_i \leftarrow L_i \cup \{S \cup \{t\} :\ (t \in V^{\setminus i} \setminus S) \wedge (b(\mathcal{L}_i, S \cup \{t\}) < \underline{\mathrm{C4}}(S \cup \{t\}))\}$ **if** $d <$ in-d.
  end for
  $d \leftarrow d + 1$
end while

For small datasets, it is feasible to score every candidate parent set so that we can compare how far the upper bounds for a given parent set S (and all its supersets) are from the true best score among itself and all supersets. Figure 1 shows such a comparison for variable *Standard-of-living-index* in the *cmc* dataset. It is clear that the new bound $\underline{\text{C4}} = \min\{\underline{g}; \underline{h}\}$ is much tighter than the current best bound in the literature (here called f) and improves considerably towards the true best score.

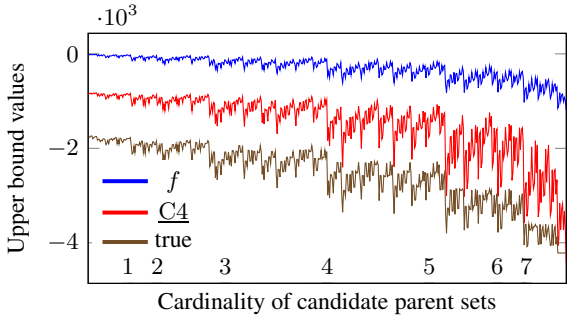

Figure 1: Upper bound values for each candidate parent set for variable Standard-of-living-index in the cmc dataset. Parent sets are arbitrarily ordered within the same cardinality.

The practical benefits of the new bounds are best observed when comparing the number of scores computed to construct $L = \bigcup_{i \in V} L_i$ for each dataset. In Figure 2, we see that the previously available bound (orange-square curves) is indeed loose as the number of scores computed is often closer to the size of the entire search space (green-diamond curves). Conversely, each of the new bounds ($\underline{g}$ and $\underline{h}$) often reduces the computational costs by more than half with respect to f's. It is also worth noticing that bound $\underline{h}$ does not always dominate $\underline{g}$, or vice versa. For instance, for datasets *zoo* and *heart-h*, $\underline{h}$ was more effective, while $\underline{g}$ was more active in the remaining datasets of Figure 2. That justifies combining $\underline{g}$ and $\underline{h}$ into $\underline{\text{C4}}$.

We also ran Algorithm 1 for the UCI datasets presented in Table 2 with the maximum in-degree as defined there. The size of the search space (for all variables in the dataset) is also shown, together with the number of pruned cases. The results in Table 2 show the number of computations pruned with bound $\underline{\text{C4}}$ is up to an order of magnitude higher in comparison to bound f. An interesting result was obtained for the *diabetes* dataset, where pruning takes places for BDeu but failed to happen for the BIC score [11], which is understood as having stronger pruning available.

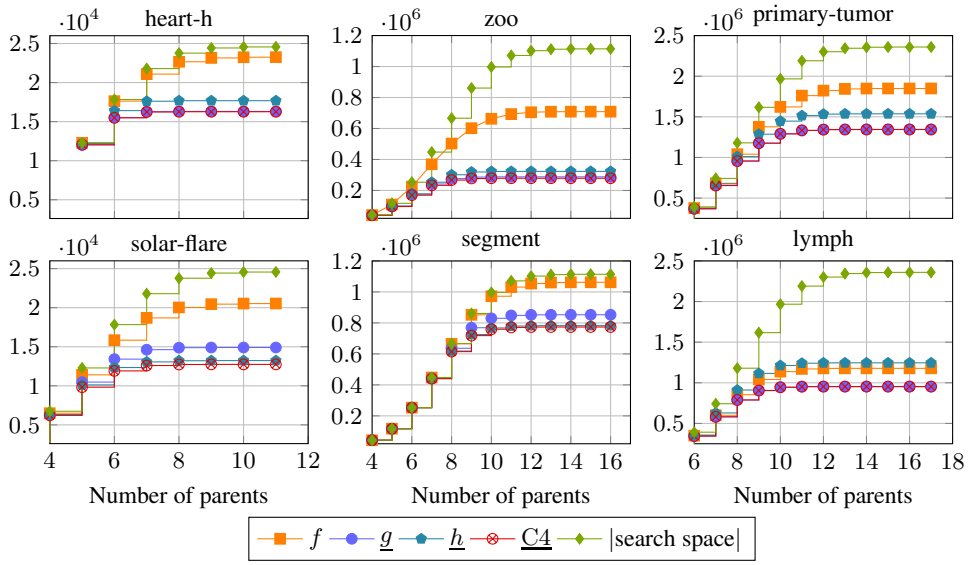

Figure 2: Number of scores computed per maximum number of parents with different bounds.

Table 2: Number of computations pruned ($|L^c| = |\text{search space}| - |L|$) with each bound: f, $\underline{g}$, $\underline{h}$ and $\underline{\text{C4}}$. Each dataset is characterised by its number of variables and observations, n and N, and the number of all possible parent combinations $|\text{search space}|$. in-d is the maximum imposed in-degree and $\frac{|\underline{g}<\underline{h}|}{|\underline{h}<\underline{g}|}$ is the ratio of the number of times bound $\underline{g}$ was active over $\underline{h}$ within $\underline{\text{C4}}$.

Dataset	n	N	$\lvert\text{search space}\rvert$	in-d	$\lvert L^c_f\rvert$	$\lvert L^c_{\underline{g}}\rvert$	$\lvert L^c_{\underline{h}}\rvert$	$\lvert L^c_{\underline{\text{C4}}}\rvert$	$\frac{\lvert\underline{g}<\underline{h}\rvert}{\lvert\underline{h}<\underline{g}\rvert}$
diabetes	9	768	2,304	5	0	0	0	0	∞
				7	0	72	184	184	123.176
				∞	0	81	193	193	123.176
nursery	9	12,960	2,304	5	0	0	342	342	6.176
				7	8	188	626	630	5.331
				∞	16	197	635	639	5.331
cmc	10	1,473	5,120	5	0	23	35	47	7.556
				7	6	746	766	828	6.045
				∞	16	846	866	928	6.045
heart-h	12	294	24,576	5	0	252	86	252	1.434
				8	1,109	7,469	6,090	7,504	1.019
				∞	1,321	8,273	6,894	8,308	1.019
solar-flare	12	1,066	24,576	5	884	1,810	2,170	2,462	2.799
				8	3,741	8,890	10,561	11,043	2.885
				∞	4,043	9,672	11,348	11,834	2.877
vowel	14	990	$1.147 \cdot 10^5$	5	1,564	1,833	1,707	1,837	0.182
				9	7,579	22,701	21,962	22,729	$6.152 \cdot 10^{-2}$
				∞	8,350	26,298	25,411	26,330	$6.162 \cdot 10^{-2}$
zoo	17	101	$1.114 \cdot 10^6$	5	7,760	18,026	18,818	20,604	0.252
				11	$3.782 \cdot 10^5$	$7.834 \cdot 10^5$	$7.483 \cdot 10^5$	$7.925 \cdot 10^5$	0.104
				∞	$4.054 \cdot 10^5$	$8.262 \cdot 10^5$	$7.91 \cdot 10^5$	$8.353 \cdot 10^5$	0.104
vote	17	435	$1.114 \cdot 10^6$	5	0	0	0	0	0.919
				11	40,544	$2.594 \cdot 10^5$	$2.126 \cdot 10^5$	$2.776 \cdot 10^5$	0.414
				∞	55,067	$3.021 \cdot 10^5$	$2.552 \cdot 10^5$	$3.203 \cdot 10^5$	0.414
segment	17	2,310	$1.114 \cdot 10^6$	5	0	0	0	0	0.184
				11	39,948	$2.229 \cdot 10^5$	$2.915 \cdot 10^5$	$2.915 \cdot 10^5$	0.256
				∞	51,902	$2.614 \cdot 10^5$	$3.317 \cdot 10^5$	$3.317 \cdot 10^5$	0.256
pendigits	17	10,992	$9.83 \cdot 10^5$	5	0	0	0	0	0
				11	0	2,386	47,757	41,619	$4.143 \cdot 10^{-2}$
				∞	0	17,445	76,982	70,321	$4.383 \cdot 10^{-2}$
lymph	18	148	$2.359 \cdot 10^6$	5	7,295	8,344	6,076	8,344	12,375.846
				11	$1.02 \cdot 10^6$	$1.237 \cdot 10^6$	$9.489 \cdot 10^5$	$1.237 \cdot 10^6$	73,280.769
				∞	$1.182 \cdot 10^6$	$1.406 \cdot 10^6$	$1.114 \cdot 10^6$	$1.406 \cdot 10^6$	73,327.538
primary-tumor	18	339	$2.359 \cdot 10^6$	5	2,460	3,555	2,667	3,555	$5.465 \cdot 10^{-2}$
				11	$4.292 \cdot 10^5$	$8.572 \cdot 10^5$	$6.751 \cdot 10^5$	$8.572 \cdot 10^5$	$6.856 \cdot 10^{-3}$
				∞	$5.105 \cdot 10^5$	$1.015 \cdot 10^6$	$8.223 \cdot 10^5$	$1.015 \cdot 10^6$	$6.797 \cdot 10^{-3}$
vehicle	19	846	$4.981 \cdot 10^6$	5	0	108	54	108	1.197
				12	$6.614 \cdot 10^5$	$2.082 \cdot 10^6$	$1.848 \cdot 10^6$	$2.12 \cdot 10^6$	0.474
				∞	$7.582 \cdot 10^5$	$2.319 \cdot 10^6$	$2.084 \cdot 10^6$	$2.358 \cdot 10^6$	0.473
hepatitis	20	155	$7.864 \cdot 10^6$	5	0	0	0	0	397.196
				12	$2.155 \cdot 10^6$	$3.341 \cdot 10^6$	$2.338 \cdot 10^6$	$3.341 \cdot 10^6$	8,198.164
				∞	$2.599 \cdot 10^6$	$3.795 \cdot 10^6$	$2.905 \cdot 10^6$	$3.795 \cdot 10^6$	6,100.199
colic	23	368	$9.647 \cdot 10^7$	5	1,170	2,415	934	2,415	∞
				14	$2.116 \cdot 10^7$	$2.122 \cdot 10^7$	$2.042 \cdot 10^7$	$2.122 \cdot 10^7$	∞
				∞	$2.277 \cdot 10^7$	$2.284 \cdot 10^7$	$2.203 \cdot 10^7$	$2.284 \cdot 10^7$	∞
autos	26	205	$8.724 \cdot 10^8$	5	$1.388 \cdot 10^5$	$1.829 \cdot 10^5$	$1.544 \cdot 10^5$	$1.829 \cdot 10^5$	∞
				15	$1.265 \cdot 10^8$	$1.265 \cdot 10^8$	$1.258 \cdot 10^8$	$1.265 \cdot 10^8$	45,904.333
				∞	$1.432 \cdot 10^8$	$1.432 \cdot 10^8$	$1.425 \cdot 10^8$	$1.432 \cdot 10^8$	45,904.333
flags	29	194	$7.785 \cdot 10^9$	5	$2.782 \cdot 10^5$	$2.834 \cdot 10^5$	$1.275 \cdot 10^5$	$2.834 \cdot 10^5$	∞
				17	$1.085 \cdot 10^9$	$1.085 \cdot 10^9$	$1.083 \cdot 10^9$	$1.085 \cdot 10^9$	∞
				∞	$1.196 \cdot 10^9$	$1.196 \cdot 10^9$	$1.194 \cdot 10^9$	$1.196 \cdot 10^9$	∞

7 Conclusions

We have devised new theoretical bounds for learning Bayesian networks with the BDeu score. These bounds come from analysing the score function from multiple angles and provide significant benefits in reducing the search space of parent sets for each node of the network. Empirical results with multiple UCI datasets illustrate the benefits that can be achieved in practice with the theoretical bounds. In particular, the new bounds allow us to explore the whole search space of parent sets using BDeu more efficiently without imposing bounds on the maximum in-degree, which was a major bottleneck before for domains beyond some dozen variables.

As future work, tighter bounds may be possible by replacing the maximum likelihood estimation terms in the formulas, as well as by using different search orders for exploring the space of parent sets, which could benefit even further from these bounds. In particular, if one would run a branch-and-bound approach to explore the parent sets of a node, it would be possible to use these bounds more effectively by not only considering the parent sets and corresponding full instantiations but also partial instantiations that are formed by disallowing some variables to be parents in some of the branches. The mathematical details to realise such ideas as well as an improved implementation of our bounds using sophisticated tailored data structures are natural next steps in this research.

References

[1] Mark Bartlett and James Cussens. Integer linear programming for the Bayesian network structure learning problem. *Artificial Intelligence*, 244:258–271, 2017.

[2] Eunice Yuh-Jie Chen, Yujia Shen, Arthur Choi, and Adnan Darwiche. Learning Bayesian networks with ancestral constraints. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29 (NIPS)*, pages 2325–2333. Curran Associates, Inc., 2016.

[3] David M. Chickering, David Heckerman, and Christopher Meek. Large-sample learning of Bayesian networks is NP-hard. *Journal of Machine Learning Research*, 20:1287–1330, October 2004.

[4] Gregory F. Cooper and Edward Herskovits. A Bayesian Method for the Induction of Probabilistic Networks from Data. *Machine Learning*, 9:309–347, 1992.

[5] James Cussens. Bayesian network learning with cutting planes. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 153–160. AUAI Press, 2011.

[6] James Cussens. An upper bound for BDeu local scores. Proc. ECAI-2012 workshop on algorithmic issues for inference in graphical models (AIGM), 2012.

[7] James Cussens and Mark Bartlett. GOBNILP 1.6.2 User/Developer Manual, 2015.

[8] James Cussens, Mark Bartlett, Elinor M. Jones, and Nuala A. Sheehan. Maximum likelihood pedigree reconstruction using integer linear programming. *Genetic Epidemiology*, 37(1):69–83, 2013.

[9] Cassio de Campos and Qiang Ji. Properties of Bayesian Dirichlet scores to learn Bayesian network structures. In *Conference on Advancements in Artificial Intelligence (AAAI)*, pages 431–436, 2010.

[10] Cassio de Campos and Qiang Ji. Efficient structure learning of Bayesian networks using constraints. *Journal of Machine Learning Research*, 12:663–689, 2011.

[11] Cassio de Campos, Mauro Scanagatta, Giorgio Corani, and Marco Zaffalon. Entropy-based pruning for learning bayesian networks using bic. *Artificial Intelligence*, 260(C):42–50, 2018.

[12] Cassio de Campos, Zhi Zeng, and Qiang Ji. Structure learning of Bayesian networks using constraints. In *Proc. of the 26th International Conference on Machine Learning (ICML)*, volume 382, pages 113–120. ACM, 2009.

[13] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

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

[15] Tommi Jaakkola, David Sontag, Amir Globerson, and Marina Meila. Learning Bayesian network structure using LP relaxations. In *Proceedings of 13th International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, volume 9, pages 358–365, 2010. Journal of Machine Learning Research Workshop and Conference Proceedings.

[16] Mikko Koivisto. Parent Assignment Is Hard for the MDL, AIC, and NML Costs. In *Computational Learning Theory (COLT)*, volume 4005, pages 289–303. Springer, 2006.

[17] Mikko Koivisto and Kismat Sood. Exact Bayesian structure discovery in Bayesian networks. *Journal of Machine Learning Research*, 5:549–573, 2004.

[18] Andrew Moore and Mary Soon Lee. Cached sufficient statistics for efficient machine learning with large datasets. *Journal of Artificial Intelligence Research*, 8:67–91, 1998.

[19] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

[20] Marc Teyssier and Daphne Koller. Ordering-based search: A simple and effective algorithm for learning Bayesian networks. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 584–590, 2005.

[21] Changhe Yuan and Brandon Malone. An improved admissible heuristic for learning optimal Bayesian networks. In *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 924–933, Catalina Island, CA, 2012.

[22] Changhe Yuan and Brandon Malone. Learning optimal Bayesian networks: A shortest path perspective. *Journal of Artificial Intelligence Research*, 48:23–65, October 2013.

Appendix A - Proofs

Lemma 1. *For $S \subseteq T \subseteq V^{\setminus i}$, $j_S \in \mathcal{D}_u^S$ and $j_T \in \mathcal{D}_u^T$ with $j_T^S = j_S$, $|\mathcal{D}_u^{T\cup\{i\}}| \geqslant |\mathcal{D}_u^{S\cup\{i\}}|$ and $|\mathcal{D}_u^{T\cup\{i\}}(j_T)| \leqslant |\mathcal{D}_u^{S\cup\{i\}}(j_S)|$.*

Proof. Given that $S \subseteq T \subseteq V^{\setminus i}$, every instantiation in $\mathcal{D}_u^{S\cup i}$ is compatible with one or more instantiations in $\mathcal{D}_u^{T\cup i}$, and it follows that $|\mathcal{D}_u^{T\cup i}| \geqslant |\mathcal{D}_u^{S\cup i}|$. The relationship is reversed when we consider the number of unique occurrences compatible with a given instantiation. By construction $j_T^S = j_S$, so if there is an instantiation $j_T \in \mathcal{D}_u^T$, there must be at least one instantiation $j_S \in \mathcal{D}_u^S$, and it follows that $|\mathcal{D}_u^{T\cup\{i\}}(j_T)| \leqslant |\mathcal{D}_u^{S\cup\{i\}}(j_S)|$. Note that both $|\mathcal{D}_u^{T\cup\{i\}}(j_T)|$ and $|\mathcal{D}_u^{S\cup\{i\}}(j_S)|$ are bounded by $q(i)$ – one instantiation for each possible value child i can assume. □

Lemma 2. *Let x be a positive integer. Then $\Gamma_\alpha(0) = 1$ and $\log\Gamma_\alpha(x) = \sum_{\ell=0}^{x-1}\log(\ell+\alpha)$.*

Proof. Follows from definition and $\Gamma(x+1) = x\Gamma(x)$. □

Lemma 3. *For x positive integer and $v \geqslant 1$, it holds that $\log\big(\Gamma_\alpha(x)/\Gamma_{\alpha/v}(x)\big) \geqslant \log v$.*

Proof. By applying Lemma 2, we obtain

$$\sum_{\ell=0}^{x-1}\log\frac{\ell+\alpha}{\ell+\alpha/v} = \log v + \sum_{\ell=1}^{x-1}\log\frac{\ell+\alpha}{\ell+\alpha/v} \geqslant \log v$$

because each term of the sum (if any) is greater than zero. □

Lemma 4. *Let x, y be non-negative integers with $x + y > 0$.*

$$\begin{cases}\Gamma_\alpha(x+y) = \Gamma_\alpha(x)\,\Gamma_\alpha(y) & \text{if } x\cdot y = 0\,,\\ \Gamma_\alpha(x+y) \geqslant \Gamma_\alpha(x)\,\Gamma_\alpha(y)\,(1+y/\alpha) & \text{otherwise.}\end{cases}$$

Proof. If either x or y are zero, then their term will cancel out and the equality holds. Otherwise we apply Lemma 2 three times and manipulate the products:

$$\begin{aligned}\frac{\Gamma_\alpha(x+y)}{\Gamma_\alpha(x)\,\Gamma_\alpha(y)} &= \frac{\prod_{z=0}^{x+y-1}(z+\alpha)}{\prod_{z=0}^{y-1}(z+\alpha)\prod_{z=0}^{x-1}(z+\alpha)}\\ &= \prod_{z=y}^{x+y-1}(z+\alpha)\prod_{z=0}^{x-1}\frac{1}{(z+\alpha)} = \prod_{z=0}^{x-1}\frac{y+z+\alpha}{z+\alpha} \geqslant \frac{y+\alpha}{\alpha}\,.\end{aligned}$$

□

Lemma 5. *For $S \subseteq V^{\setminus i}$ and $j \in \mathcal{D}_u^S$, assume that $\vec{n}_j = (n_{j,k})_{k\in c(i)}$ are in decreasing order over $k = 1,\ldots,q(i)$ (this is without loss of generality, since we can name and process them in any order). Then for any $\alpha \geqslant \alpha_j = \alpha_{ess}/q(S)$, we have*

$$\mathrm{LLBDeu}(S,j) \leqslant f(S,j) + g(S,j,\alpha),\ \textit{with}\ g(S,j,\alpha) = -\sum_{l=1}^{k'-1}\log\left(1+n_{j,l}/\alpha\right),$$

and $k' \leqslant k$ is the largest index such that $n_{j,k'} > 0$.

Proof. Since counts $n_{j,k}$ are in decreasing order by k, we apply Corollary 1:

$$\mathrm{LLBDeu}(S,j) \leqslant -\log\left(\prod_{l=1}^{q(i)}\Gamma_{\alpha_j}(n_{j,l})\prod_{l=1}^{k'-1}(1+\frac{n_{j,l}}{\alpha_j})\right) + \sum_{k\in c(i)}\log\Gamma_{\alpha_{j,k}}(n_{j,k}) =$$

$$\sum_{k\in c(i)}\log\left(\frac{\Gamma_{\alpha_{j,k}}(n_{j,k})}{\Gamma_{\alpha_j}(n_{j,k})}\right) - \sum_{l=1}^{k'-1}\log\left(1+\frac{n_{j,l}}{\alpha_j}\right) \leqslant -|\mathcal{D}_u^{S\cup\{i\}}(j)|\log q(i) - \sum_{l=1}^{k'-1}\log\left(1+\frac{n_{j,l}}{\alpha}\right),$$

with $\alpha \geqslant \alpha_j$ and $\Gamma_{\alpha_{j,k}}(n_{j,k})/\Gamma_{\alpha_j}(n_{j,k}) \leqslant -\log q(i)$ by Lemma 3 whenever $n_{j,k} > 0$. □

Lemma 6. *For $S \subseteq T \subseteq V^{\setminus i}$, $j_T \in \mathcal{D}_u^T$, and $j_S \in \mathcal{D}_u^S$ with $j_T^S = j_S$, $f(T, j_T) \geqslant f(S, j_S)$ and $g(T, j_T, \alpha) \geqslant g(S, j_S, \alpha)$.*

Proof. Because $j_T^S = j_S$, $|\mathcal{D}_u^{T\cup\{i\}}(j_T)| \leqslant |\mathcal{D}_u^{S\cup\{i\}}(j_S)|$. Moreover, $n_{j_T,k} \leqslant n_{j_S,k}$ for every $k \in c(i)$ (the counts get partitioned as more parents are introduced to arrive at T from S), so $(1 + n_{j_T,k}/\alpha) \leqslant (1 + n_{j_S,k}/\alpha)$ for every k, and the result follows. □

Lemma 7. *Let $S \subseteq V^{\setminus i}$ and $j \in \mathcal{D}_u^S$. Then* $\mathrm{LLBDeu}(S, j) \leqslant ML(\vec{n}_j)$, *where we have that* $ML(\vec{n}_j) = \sum_{k\in c(i)} n_{j,k} \log(n_{j,k}/n_j)$. *($0 \log 0 = 0$.)*

Proof. The llB is simply the log probability of observing a data sequence with counts $\vec{n}_j$ under a Dirichlet-multinomial distribution with parameter vector $\vec{\alpha}_j$. The result follows from Expression (1) and holds for any prior $\vec{\alpha}_j$. □

Lemma 8. *If $\nexists k : n_{j,k} = n_j$, then $h_{\vec{n}_j}$ is a concave function for positive $\alpha \leqslant 1$.*

Proof. Using the identity in Lemma 2, or, equivalently, by exploiting known properties of the digamma and trigamma functions we have:

$$\frac{\partial h_{\vec{n}_j}}{\partial \alpha}(\alpha) = \sum_{k=1}^{q(i)} \sum_{\ell=0}^{n_{j,k}-1} \frac{1}{\ell q(i) + \alpha} - \sum_{\ell=0}^{n_j-1} \frac{1}{\ell + \alpha}$$

and

$$\frac{\partial^2 h_{\vec{n}_j}}{\partial \alpha^2}(\alpha) = \sum_{\ell=0}^{n_j-1} \frac{1}{(\ell + \alpha)^2} - \sum_{k=1}^{q(i)} \sum_{\ell=0}^{n_{j,k}-1} \frac{1}{(\ell q(i) + \alpha)^2}\,.$$

It suffices to show that $\frac{\partial^2 h_{\vec{n}_j}}{\partial \alpha^2}(\alpha)$ is always negative under the conditions of the theorem. If there are at least two $n_{j,k} > 0$, then

$$\frac{\partial^2 h_{\vec{n}_j}}{\partial \alpha^2}(\alpha) \leqslant \sum_{\ell=0}^{n_j-1} \frac{1}{(\ell + \alpha)^2} - \frac{2}{\alpha^2}$$

simply by ignoring all those negative terms with $\ell \geqslant 1$.

Now we approximate it by the infinite sum of quadratic reciprocals:

$$\begin{aligned}\frac{\partial^2 h_{\vec{n}_j}}{\partial \alpha^2}(\alpha) &\leqslant \sum_{\ell=0}^{n_j-1} \frac{1}{(\ell + \alpha)^2} - \frac{2}{\alpha^2} = -\frac{1}{\alpha^2} + \frac{1}{(1+\alpha)^2} + \sum_{\ell=2}^{n_j-1} \frac{1}{(\ell + \alpha)^2} \\ &< -\frac{1}{\alpha^2} + \frac{1}{(1+\alpha)^2} + \sum_{\ell=2}^{\infty} \frac{1}{\ell^2} = -\frac{1}{\alpha^2} + \frac{1}{(1+\alpha)^2} + \frac{\pi^2}{6} - 1\,,\end{aligned}$$

which is negative for any $\alpha \leqslant 1$ (the gap between the two fractions containing α obviously decreases with the increase of α, so it is enough to check the sign for the largest value $\alpha = 1$). Thus we have $\frac{\partial^2 h_{\vec{n}_j}}{\partial \alpha^2}(\alpha) < 0$. □

Lemma 9. *Let $S \subseteq V^{\setminus i}$ and $j \in \mathcal{D}_u^{V^{\setminus i}}$ such that $\nexists k : n_{j,k} = n_j$. If $\alpha \leqslant q(S)$ and $\frac{\partial h_{\vec{n}_j}}{\partial \alpha}(\alpha/q(S))$ is non-negative then $h_{\vec{n}_j}(\alpha/q(T)) \leqslant h_{\vec{n}_j}(\alpha/q(S))$ for every $T \supseteq S$.*

Proof. Since $\nexists k : n_{j,k} = n_j$ and $\alpha/q(S) \leqslant 1$, we have that $h_{\vec{n}_j}$ is concave (Lemma 8) and since $\frac{\partial h_{\vec{n}_j}}{\partial \alpha}(\alpha/q(S)) \geqslant 0$, $h_{\vec{n}_j}$ is non-decreasing. □

Corollary 1. *Let $x_1, \ldots, x_k$ be a list of non-negative integers in decreasing order with $x_1 > 0$, then*

$$\Gamma_\alpha\left(\sum_{l=1}^{k} x_l\right) \geqslant \prod_{l=1}^{k} \Gamma_\alpha\left(x_l\right) \prod_{l=1}^{k'-1} (1 + x_l/\alpha)\,,$$

where $k' \leqslant k$ is the last positive integer in the list (the second product disappears if $k' = 1$).

Proof. Repeatedly apply Lemma 4 to $x_t + (\sum_{l=t}^{k} x_l)$ until all elements are processed. While both the current x_t and the rest of the list are positive (that is, until $t = k' - 1$), we gain the extra term $(1 + x_t/\alpha)$. After that, we only 'collect' the Gamma functions, so the result follows. □

Corollary 2. *Let* $S \subseteq V^{\setminus i}$ *and* $j_S \in \mathcal{D}_u^S$. *Then* $\mathrm{LLBDeu}(S, j_S) \leqslant \sum_{j \in \mathcal{D}_u^{V^{\setminus i}} :\, j^S = j_S} \mathit{ML}(\vec{n}_j)$.

Proof. This follows from the properties of the maximum likelihood estimation, because it is monotonically non-decreasing with the expansion of parent sets (we fit better in maximum likelihood when having more parents). □