人工知能学会研究会資料
SGI-FPAI-B503-16

# Efficient Bayesian Network Structure Learning for Maximizing the Posterior Probability

Joe Suzuki[1] *

[1] Graduate School of Science, Osaka University

**Abstract:** This paper addresses the problem of efficiently finding an optimal Bayesian network structure for maximizing the posterior probability. In particular, we focus on the B& B strategy to save the computational effort associated with finding the largest score. To make the search more efficient, we need a tighter upper bound so that the current score can exceed it more easily. We find two upper bounds and prove that they are tighter than the existing one (Campos and Ji, 2011). Finally, we demonstrate that the proposed two bounds render the search to be much more efficient using the Alarm and other major data sets. For example, the search is three to four times faster for $n = 100$ and two to three times faster for $n = 500$. We also experimentally verify that the overhead due to replacing the existing search by the proposed one is negligible.

## 1 Introduction

In this paper, we explore the conditional independence relations among variables from $n$ samples w.r.t. $N$ variables and express them using a Bayesian network (BN) [15].

One of the reasons that BNs are not currently used much in real applications is that construction of a BN is computationally hard when $N$ is large. The main topic of this paper involves reduction of the computational effort.

There are many approaches to BN structure learning (BNSL). Some execute conditional independence tests for each triple of exclusive sets to estimate an undirected graph before deciding the orientations of the edges. For example, the PC algorithm [14] belongs to this class of approaches. Others calculate a local score for each subset $S$ of the whole variable set $\{X_1, \cdots, X_N\}$ to find a factorization that maximizes the global score, which can be obtained from the associated local scores. It is assumed that the optimal structure maximizes either the posterior probability or its variant.

In this paper, we assume prior probabilities over the structures and the parameters to marginalize the parameter values and to choose a structure that maximizes the posterior probability given the data (Cooper and Herskovits 1991 [6]).

Suzuki in 1993 [16] applied the minimum description length (MDL) principle [11] to the BNSL problem. Suppose that the data can be described via a rule and its exceptions in many ways. The MDL principle chooses the rule that minimizes the total length. However, minimizing the description length does not mean maximizing the posterior probability. The former is merely an approximation of the latter, although both estimate the correct structure for large $n$ (consistency). The merit of applying the former over the latter is that the branch and bound (B&B) technique can be applied to the former. If the B&B rule works effectively, insufficient candidates will be excluded without computation and optimal solutions will be found in a shorter period of time because the search space will be limited due to exclusion of unnecessary computational effort. The B&B for the MDL was first proposed by Suzuki, and the idea was used by many authors such as Tian in 2000 [20] and Campos and Ji in 2011 [7].

Recently, a similar B& B technique was found for maximizing the posterior probability (Campos and Ji 2011 [7]). This idea is applied to finding the optimal parent sets in the BNSL problem. Suppose that we search a parent set of $X_i$ with the largest score and that for each candidate $S \subseteq \{X_1, \cdots, X_{i-1}, X_{i+1}, \cdots, X_N\}$, we compute an upper bound $UB(S)$ of the score for all $T$ ($\supseteq S$). If the score of $S \backslash \{Y\}$ of any $Y \in S$ exceeds $UB(S)$, then we can exclude the possibility that $T$ ($\supseteq S$) is optimal.

The upper bound was derived by Campos and Ji in 2011 [7] for the BDeu prior model ([3, 21]) and was extended to a wider class [19], although the derivations are essentially the same. The main drawback of both methods is that the bounds [7, 19] are still loose, which can be investigated using experiments so that the bound does not require frequent unnecessary computational effort.

Our contributions in this paper are the following:

1. Deriving upper bounds that are tighter than the existing ones
2. Demonstrating that the new bounds are significantly more effective than the existing ones, particularly for small sample sizes, using the Alarm [2] and other major datasets (the first $n$ samples and the first $N$ variables).

*連絡先：大阪大学大学院理学研究科
〒 560-0043 豊中市待兼山町 1-1
E-mail: suzuki@math.sci.osaka-u.ac.jp

The tightest proposed bound resulted in the computational time being three to four times faster for $n = 100$ and two to three times faster for $n = 500$. We also verified that the overhead due to replacing the existing bound by the proposed one is negligible.

This paper is organized as follows: Section 2 provides a background for understanding the results of this paper. Sections 2.1, 2.2, 2.3, and 2.4 explain the BN, BNSL, dynamic programming approach, and B&B approach, respectively. Section 3 proposes two upper bounds and proves that the proposed upper bounds are tighter than the existing ones. Section 4 demonstrates the upper bounds and compares them with the existing ones in terms of the efficiencies. Section 5 summarizes the results and discusses future work.

# 2 Preliminaries

In this section, we provide basic materials to understand the results of this paper.

## 2.1 Bayesian Network

Let $X_1, \cdots, X_N$ $(N \geq 1)$ be random variables that take on a finite number of values. We define a Bayesian network (BN) using an directed acyclic graph (DAG) that expresses the factorization of the distribution

$$P(X_1, \cdots, X_N) = P(X_1|\Pi_1) \cdots P(X_N|\Pi_N) \ ,$$

where $\Pi_i$ is a subset of $\{X_1, \cdots, X_{i-1}, X_{i+1}, \cdots, X_N\}$, hereafter referred to as a parent set of $X_i$. If we direct an edge from each element of $\Pi_i$ to $X_i$ for $i = 1, \cdots, n$, there should be no directed loop among the directed edges.

Each BN is expressed by such a structure and the conditional probabilities $(P(X_i = x|\Pi_i = y))_{x,y}$ for each $i = 1, \cdots, N$, where $\Pi_i$ is a subset of the variable set $\{X_1, \cdots, X_{i-1}, X_{i+1}, \cdots, X_N\}$ and $\Pi_i = y$ implies that $\Pi_i$ is one of the combinations. For example, if $X_i = X$, $\Pi_i = \{Y, Z\}$, and $Y$ and $Z$ lie in the finite sets $\mathcal{Y}$ and $\mathcal{Z}$, respectively, then $\Pi_i = y$ means that $y$ is in the finite set $\mathcal{Y} \times \mathcal{Z}$.

## 2.2 Learning Bayesian network structures

Given $n$ tuples of examples

$$\begin{array}{ccccc} X_1 = x_{1,1} & , \cdots , & X_N = x_{1,N} \\ \cdots , & \cdots , & \cdots \\ X_1 = x_{n,1} & , \cdots , & X_N = x_{n,N} \end{array}$$

of variables $X_1 \cdots, X_N$, we learn the BN structure. The problem is to choose a DAG of the $N$ variables that maximizes the posterior probability given the data. There are two prior probabilities for BNs: over the structures and over the parameters. We assume the following:

1. There are no missing values in the $n$ tuples of examples
2. The prior probabilities over the structures are uniform.

We construct local scores $Q^n(U)$ for each subset $U$ of the whole variable set $V$ by marginalizing the unknown parameter values based on the prior probability.

Suppose that $(X, Y) = (x, y)$ occurs $c(x, y)$ times in $(x_1, y_1, \cdots, x_n, y_n)$. Then, if we are given the prior probability $w(\theta)$ over the parameters $\theta = (\theta(x, y))$, the conditional score of $X$ given $Y$ can be expressed by

$$Q^n(X|Y) := \int \prod_x \prod_y \theta(x, y)^{c(x,y)} w(\theta) d\theta$$

We assume that the priors over the parameters are expressed by

$$w(\theta) = K \prod_x \prod_y \theta(x, y)^{a(x,y)-1} \ ,$$

where $a(x, y) > 0$ is a constant and $K$ is a normalizing constant. Then, the local score of $(y_1, \cdots, y_n)$ given $(x_1, \cdots, x_n)$ can be expressed by [9]

$$\begin{aligned} & Q^n(X|Y) \\ = & \int \prod_x \prod_y \theta(x, y)^{c(x,y)} w(\theta) d\theta \qquad (1) \\ = & \prod_y \frac{\Gamma(\sum_x a(x, y)) \prod_x \Gamma(c(x, y) + a(x, y))}{\prod_x \Gamma(a(x, y)) \Gamma(c(y) + a(y))} \qquad (2) \end{aligned}$$

where $c(y) = \sum_x c(x, y)$ and $a(y) = \sum_x a(x, y)$.

In a similar way, we construct $Q^n(X|Z)$ and $Q^n(X|YZ)$, as well as $Q^n(Y|\cdot)$ and $Q^n(Z|\cdot)$. We can check that $Q^n(X|Y)Q^n(Y) = Q^n(X)Q^n(Y|X)$, but we do not prove it.

Thus far, we have illustrated the $N = 3$ case. However, for general $N$, the computation has been already proven to be NP-hard [5].

## 2.3 Dynamic Programming Approach

In 2006, Silander and Myllymaki [12][1] proposed a way of efficiently finding a BN structure with the maximum posterior probability. The procedure consists of two phases.

In the first phase, for each $X \in V$ and $S \subseteq V \backslash \{X\}$, we compute the subset $U \subseteq S$ that maximizes $Q^n(X|U)$, referred to as $\pi_S(X)$. For the $N = 3$ case, if $S = \{\}$, then we do not have to compare anything; if $S = \{Y\}$, then we compare $Q^n(X)$ and $Q^n(X|Y)$; if $S = \{Z\}$, then we compare $Q^n(X)$ and $Q^n(X|Z)$; and if $S = \{Y, Z\}$, then we compare $Q^n(X)$, $Q^n(X|Y)$, $Q^n(X|Z)$, and $Q^n(X|YZ)$.

[1] The idea of using dynamic programing was proposed by A. P. Singh & A. W. Moore (2005)

In the second phase, we seek a permutation $X(1), \cdots, X(N)$ of $X_1, \cdots, X_N$ such that $S(1) := \{\}$, $S(i+1) := S(i) \cup \{X(i)\}$, $i = 1, \cdots, N-1$ and the product

$$\prod_{i=1}^{N} Q^n(X(i)|\pi_{S(i)}(X(i)))$$

is maximized.

In this paper, we mainly consider the first phase. However, we need to compute $N2^{N-1}$ conditional local scores $Q^n(X|S)$ to obtain $\pi_S(X)$ for $X \in V$ and $S \subseteq V\backslash\{X\}$. Without an exponentially large computational effort, it is hard to learn the BN structure from the data.

## 2.4 Branch & Bound Approach

The main idea (Campos and Ji [7]) is that for $Y \in S \subseteq V\backslash\{X\}$, if

$$Q^n(X|\pi_{S\backslash\{Y\}}(X)) \geq \sup_{T \supseteq S} Q^n(X|\pi_T(X)) \ ,$$

then we do not have to compute $Q^n(X|T)$ for $T \supseteq S$. For example, in Figure **??**, if we know

$$Q^n(X|\pi_Z(X)) \geq \sup_{T \supseteq \{Y,Z\}} Q^n(X|T) \ ,$$

in some way, then from $Q^n(X|\pi_Z(X)) \geq Q^n(X|Y,Z)$, in order to obtain $Q^n(X|\pi_{YZ}(X))$, we only compare $Q^n(X|\pi_Y(X))$ and $Q^n(X|\pi_Z(X))$. Similarly, from $Q^n(X|\pi_Z(X)) \geq Q^n(X|Y,Z,W)$, in order to obtain $Q^n(X|\pi_{YZW}(X))$, we only compare $Q^n(X|\pi_{YZ}(X))$, $Q^n(X|\pi_{ZW}(X))$, and $Q^n(X|\pi_{WY}(X))$.

We execute the procedure from $S = \{\}$ to $S = V\backslash\{X\}$ in a bottom-up manner:

1. Find $p := \max_{Y \in S} Q^n(X|\pi_{S\backslash\{Y\}}(X))$;
2. compute the upper bound $q := \sup_{T \supseteq S} Q^n(X|T)$;
3. if $p \geq q$, then exclude the computations of $Q^n(X|T)$ for all $T \supseteq S$ in the future
4. else, compute $Q^n(X|S)$ and compare it with $p$,

where the original procedure consists of the first and last steps. In order to realize the third step of the modified procedure, we prepare flags to show that $Q^n(X|S)$ is not needed to compute the node $S$. Even if the flag is on for $T$, if the value of $Q^n(X|\pi_T(X))$ is improved, it will be replaced.

In fact, it is time-consuming to obtain the values of $Q^n(X|S)$ for each $X$ and $S \subseteq V\backslash\{X\}$.

Suppose that $X_i = x$ and $\Pi_i = y$ are $1, \cdots, \alpha_i$ and $1, \cdots, \beta_i$, respectively, for $i = 1, \cdots, N$. We say that the prior over the parameters is BDeu (Bayesian Dirichlet equivalent uniform [3],[21]) if $\frac{a_i(x,y)}{\alpha_i \beta_i}$ is constant for all $x = 1, \cdots, \alpha_i$, $y = 1, \cdots, \beta_i$, and $i = 1, \cdots, N$, where $c_i(y) = \sum_x c_i(x,y)$ and $a_i(y) = \sum_x a_i(x,y)$.

$$Q^n(X_i|\Pi_i) = \prod_y \{\frac{\Gamma(a_i(y))}{\Gamma(c_i(y) + a_i(y))} \cdot \prod_x \frac{\Gamma(c_i(x,y) + a_i(x,y))}{\Gamma(a_i(x,y))}\} \quad (3)$$

Campos and Ji [7] and J. Cussen [4] derived the following for the BDeu priors

$$\sup_{T \supseteq \Pi_i} Q^n(X_i|T) \leq \alpha_i^{c_i} \quad (4)$$

where $c_i$ is the number of states $\Pi_i = y$ that actually occurred in the $n$ tuples of examples ($c_i(y) \geq 1$).

We say that $\Pi_i'$ is a refinement of $\Pi_i$ if $\Pi_i$ is a subset of $\Pi_i'$ when we consider them as sets of variables on which $X_i$ depends. For example, suppose that $X_i = X$, $\Pi_i = \{Y\}$, and $\Pi_i' = \{Y, Z\}$. Then, $\Pi_i'$ is a refinement of $\Pi_i$.

Let $\Pi_i'$ be a refinement of another parent set $\Pi_i$. Suppose

$$Q^n(X_i|\Pi_i') = \prod_{y'} \{\frac{\Gamma(a_i'(y'))}{\Gamma(c_i'(y') + a_i'(y'))} \cdot \prod_x \frac{\Gamma(c_i'(x,y') + a_i'(x,y'))}{\Gamma(a_i'(x,y'))}\} \ , \quad (5)$$

where $c_i'(y') = \sum_x c_i'(x,y')$ and $a_i'(y') = \sum_x a_i'(x,y')$. Compare (3) and (5); if $\Pi_i' \in \{1, \cdots, \beta\} \Longleftrightarrow \Pi_i = y$, we have

$$c_i(x,y) = \sum_{y'=1}^{\beta} c_i'(x,y') \ . \quad (6)$$

We say that the refinements are weakly and strongly regular if

$$\frac{a_i(x,y)}{a_i(y)} = \frac{a_i'(x,y')}{a_i'(y')} \quad (7)$$

and if

$$a_i(x,y) = a_i'(x,y') \quad (8)$$

respectively, for all $x = 1, \cdots, \alpha_i$ and $y' = 1, \cdots, \beta$. The conditions (7) and (8) are satisfied by the BDeu models [3, 21] and by the Jeffreys rule [8] $a_i(x,y) = 0.5$ for all $x, y$ and $i = 1, \cdots, N$, respectively.

J. Suzuki [19] generalized the upper bound for the general BD priors under weak regularity of the refinements:

$$\sup_{T \supseteq \Pi_i} Q^n(X_i|T) \leq Q_*^n(X_i|\Pi_i) := \prod_{y:c_i(y)>0} \frac{\max_x a_i(x,y)}{a_i(y)} \ . \quad (9)$$

Note that (4) and (9) coincide for the BDeu because $a_i(x,y)/a_i(y)$ is constant and $\frac{\max_x a_i(x,y)}{a_i(y)} = \frac{1}{\alpha_i}$ for all $x$, $y$, and $i = 1, \cdots, N$.

# 3 Improved Bounds

In this section, assuming strong regularity (8) of the refinements, we derive tighter upper bounds. In particular, we construct bounds $Q_{**}^n(X_i|\Pi_i)$ and $Q_{***}^n(X_i|\Pi_i)$ such that

$$\sup_{T \supseteq \Pi_i} Q^n(X_i|T) \leq Q_{***}^n(X_i|\Pi_i) \leq Q_{**}^n(X_i|\Pi_i) \leq Q_*^n(X_i|\Pi_i) \ .$$

Although the Gamma function $\Gamma(\cdot)$ looks complicated, it simply defines a generalized version of the factorization. For example, using the formula $\Gamma(u+1) = u\Gamma(u)$ for $u > 0$ and $\Gamma(1) = 1$, we have

$$\begin{aligned}\Gamma(n) &= (n-1)\Gamma(n-1) \\ &= \cdots = (n-1)(n-2)\cdots 1 \cdot \Gamma(1) = (n-1)!\ ,\end{aligned}$$

and

$$\begin{aligned}\frac{\Gamma(n+a)}{\Gamma(a)} &= \frac{(n+a-1)\Gamma(n+a-1)}{\Gamma(a)} \\ &= \cdots = (n+a-1)\cdots a\end{aligned}$$

for integer $n \geq 0$ and real $a > 0$.

### 3.1 A Slightly Better Bound

The first bound $Q^n_{**}(X_i|\Pi_i)$ is given as

$$Q^n_{**}(X_i|\Pi_i) := \prod_y \{\frac{\Gamma(c_i(y) + \max_x a_i(x,y))}{\Gamma(\max_x a_i(x,y))} \cdot \frac{\Gamma(a_i(y))}{\Gamma(c_i(y)+a_i(y))}\}\ .$$

**Theorem 1** Under strong regularity (8) of the refinements,

$$\sup_{T \supseteq \Pi_i} Q^n(X_i|T) \leq Q^n_{**}(X_i|\Pi_i)\ .$$

We claim that the bound $Q^n_{**}(X_i|\Pi_i)$ is better than the existing bounds (6)(9):

**Theorem 2**

$$Q^n_{**}(X_i|\Pi_i) \leq Q^n_*(X_i|\Pi_i)$$

### 3.2 A Significantly Better Bound

The second bound $Q^n_{***}(X_i|\Pi_i)$ is given as

$$\begin{aligned}&Q^n_{***}(X_i|\Pi_i) \\ :=\ &\prod_x \prod_y \{\frac{\Gamma(c_i(x,y)+a_i(x,y))}{\Gamma(a_i(x,y))} \cdot \frac{\Gamma(a_i(y))}{\Gamma(c_i(x,y)+a_i(y))}\}\ .\end{aligned}$$

**Theorem 3** Under strong regularity (8) of the refinements,

$$\sup_{T \supseteq \Pi_i} Q^n(X_i|T) \leq Q^n_{***}(X_i|\Pi_i)\ .$$

Moreover, we claim that the bound $Q^n_{***}(X_i|\Pi_i)$ is better than another bound $Q^n_{**}(X_i|\Pi_i)$:

**Theorem 4**

$$Q^n_{***}(X_i|\Pi_i) \leq Q^n_{**}(X_i|\Pi_i)$$

Proof. By replacing $c(z)$ and $a(z)$ in Lemma 3 by $c_i(x,y)$ and $a_i(x,y)$, respectively, we have

$$\begin{aligned}&\prod_x \{\frac{\Gamma(c_i(x,y)+a_i(x,y))}{\Gamma(a_i(x,y))} \cdot \frac{\Gamma(a_i(y))}{\Gamma(c_i(x,y)+a_i(y))}\} \\ \leq\ &\frac{\Gamma(c_i(y)+\max_x a_i(x,y))}{\Gamma(\max_x a_i(x,y))} \cdot \frac{\Gamma(a_i(y))}{\Gamma(c_i(y)+a_i(y))}\ ,\end{aligned}$$

where $c_i(y) = \sum_x c_i(x,y)$ and $a_i(y) = \sum_x a_i(x,y)$ have been applied. By multiplying the inequality over all $y$, we obtain the theorem.

Finally, we note that Theorems 3 and 4 imply Theorem 1.

## 4 Experiments

We executed the structure learning procedures that save computational effort using the upper bounds $Q^n_*$, $Q^n_{**}$, and $Q^n_{***}$ for the case $a_i(x,y) = 0.5$. While Campos and Ji [7] claimed that the bound is good for BDeu, we can check that it is good for the case $a_i(x,y) = 0.5$ as well and that (4) and (9) coincide.

In the previous section, we have shown that the proposed bounds $Q^n_{**}$ and $Q^n_{***}$ are tighter than the existing one, $Q^n_*$. However, we need to determine when the former significantly outperforms the latter. Because the same structure that maximizes the posterior probability is obtained even if the upper bounds are different, we evaluate the performance in terms of its efficiency, i.e.,

1. the number $R$ ($\leq 2^{N-1}$) of subsets $S \subseteq V\backslash\{X\}$ for which the values of $Q^n(X|S)$ were actually computed
2. the execution time $T$ (seconds) for completing the task (finding $\pi_S(X)$ for some $X \in V$ and all $S \subseteq V\backslash\{X\}$)

for each pair of the variable size $N$ and sample size $n$. We examined the procedures for the Alarm data set (37 variables and 20,000 samples) [2]. We utilized an Intel core M-5Y10c processor and the Windows 10 operating system.

We first focused on the case with large $N$ and small $n$. In particular, we measured the values of $R$ and $T$ for $N = 16, 18, 20, 22, 24$ (the first $N$ variables out of 37) and $n = 100, 300, 500$ (the first $n$ samples out of 20,000) using $Q^n_*$, $Q^n_{**}$, and $Q^n_{***}$, where one variable (the seventh variable) is used as $X_i$ and $S$ ranges over the $2^{N-1}$ subsets. In Table 1, we write them as $R_*$, $T_*$, $R_{**}$, $T_{**}$, $R_{***}$, $T_{***}$, and $t$ denotes the actual execution time (seconds) when no branch & bound cut was applied.

From Table 1, we see that for the small sample size ($n = 100$), the proposed procedures using $Q_{**}$ and $Q_{***}$ ran around 2.2-2.4 times and 3.2-4.1 times faster than $Q_*$, respectively, for both quantities $R$ and $T$. On the other hand, for the large sample size ($n = 500$), we find that $Q_{**}$ and $Q_{***}$ were 1.4-1.8 times and 1.7-2.5 times faster than $Q^*$, respectively.

Table 1: The values of $R$ and $T$: the proposed procedures are much more efficient than the existing ones.

| (a) $n = 100$ | | | | |
|---|---|---|---|---|
| $N$ | $2^{N-1}$<br>$t$ | $R_*$<br>$T_*$ | $R_{**}$<br>$T_{**}$ | $R_{***}$<br>$T_{***}$ |
| 16 | 32,768<br>3.123 | 11,834<br>2.156 | 5,274<br>1.438 | 3,366<br>0.828 |
| 18 | 13,1072<br>11.641 | 41,903<br>3.766 | 18,143<br>2.406 | 11,342<br>1.385 |
| 20 | 524,288<br>56.875 | 145,374<br>15.047 | 61,079<br>7.562 | 37,306<br>5.063 |
| 22 | 2,097,152<br>203.766 | 514,300<br>49.843 | 212,374<br>23.954 | 127,562<br>15.406 |
| 24 | 8,388,608<br>1012.234 | 1,698,040<br>211.656 | 694,782<br>105.375 | 411,466<br>63.95 |

| (b) $n = 500$ | | | | |
|---|---|---|---|---|
| $N$ | $2^{N-1}$<br>$t$ | $R_*$<br>$T_*$ | $R_{**}$<br>$T_{**}$ | $R_{***}$<br>$T_{***}$ |
| 16 | 32,768<br>22.469 | 24,788<br>17.891 | 17,337<br>13.312 | 14,070<br>11.234 |
| 18 | 131,072<br>92.047 | 119,828<br>83.080 | 76,698<br>59.172 | 44,462<br>34.094 |
| 20 | 524,288<br>782.72 | 355,379<br>548.219 | 192,798<br>367.969 | 137,625<br>281.703 |
| 22 | 2,097,152<br>2992.078 | 1,251,530<br>1816.0630 | 705,208<br>1122.531 | 495,162<br>768.234 |
| 24 | 8,388,608<br>10720.192 | 4,380,355<br>6356.221 | 2,468,228<br>3928.858 | 1,733,067<br>2688.819 |

Comparing the procedure without B&B, $Q_{***}$ took only 5-10% of the time for $n = 100$. However, the proposed $Q_{***}$ took around 30-50% for $n = 500$, which does not suggest that $Q_{***}$ is disadvantageous because $Q_*$ took around 60-70% of the time executed by the procedure without B&B for $n = 500$. We find that even when $Q_*$ is not efficient, $Q_{***}$ works efficiently. Generally speaking, the lower the sample size $n$, the more efficient the procedure with B&B, because the Bayes method tends to avoid inferring complicated statistical models from a small number of samples when searching the maximum posterior probability model; eventually, the search space will be limited. Such a phenomenon was observed in structure learning with B&B based on the MDL principle [17]. We also see that the B&B works more efficiently for larger variable sizes of $N$, although the total time grows almost exponentially with $N$, which can be seen in any structure learning method with B&B.

To consider B&B for saving computational effort, we need to evaluate the execution time required for computing the upper bound itself. If it is too large, even if the search space is reduced, the total execution time might increase. To examine how large the overhead is, we calculated the ratios $R/2^{N-1}$ and $T/t$ for each of $Q^n_*, Q^n_{**}$, and $Q^n_{***}$ for $n = 300$, as shown in Table 2. $R_*/2^{N-1}$ is slightly lower than $T_*/t$ while $R_{**}/2^{N-1}$ and $R_{***}/2^{N-1}$ are slightly larger than $T_{**}/t$ and $T_{***}/t$, respectively. In this sense, the proposed bounds $Q^n_{**}$ and $Q^n_{***}$ require more computational effort. However, from Table 2, we can see that the overhead is negligible compared with the obtained efficiency gains using $Q^n_{**}$ and $Q^n_{***}$ rather than using $Q^n_*$. Moreover, we did not use any extra memory to replace the existing bound by the proposed ones.

Table 2: The values of $R/2^{N-1}$ and $T/t$: the overhead of the proposed upper bound is negligible.

| N | $R_*/2^{N-1}$<br>$T_*/t$ | $R_{**}/2^{N-1}$<br>$T_{**}/t$ | $R_{**}/2^{N-1}$<br>$T_{**}/t$ |
|---|---|---|---|
| 16 | 0.863<br>0.845 | 0.586<br>0.618 | 0.380<br>0.410 |
| 18 | 0.920<br>0.936 | 0.612<br>0.713 | 0.330<br>0.440 |
| 20 | 0.764<br>0.762 | 0.426<br>0.490 | 0.224<br>0.256 |
| 22 | 0.476<br>0.535 | 0.246<br>0.302 | 0.164<br>0.204 |
| 24 | 0.422<br>0.395 | 0.212<br>0.241 | 0.140<br>0.148 |

Table 3: The ratios $R_*/2^{N-1}$, $R_* * /2^{N-1}$, and $R_{***}/2^{N-1}$ for datasets insurance and adult: for insurance dataset (27 variables and 20000 samples), the first 20 variables and 100/500 instances were used for the test. For adult dataset (14 variables and 20000 samples), the first 100/500 instances were used for the test.

| dataset<br>$(N, n)$ | $R_*/2^{N-1}$ | $R_{**}/2^{N-1}$ | $R_{***}/2^{N-1}$ |
|---|---|---|---|
| Insurance<br>$(20, 100)$ | 0.626 | 0.429 | 0.298 |
| Insurance<br>$(20, 100)$ | 0.782 | 0.634 | 0.419 |
| Adult<br>$(14, 100)$ | 0.511 | 0.410 | 0.271 |
| Adult<br>$(14, 500)$ | 0.627 | 0.521 | 0.430 |

For other datasets, as seen from Table 3 similar tendencies can be observed for Insurance and Adult datasets although the differences are not so large as for the Alarm database for the Insurance. For the experiment, we evaluate the ratio $R/2^{N-1}$ for $Q_*$,$Q_{**}$, and $Q_{***}$.

Thus, we conclude that the proposed $Q_{**}$ and $Q_{***}$ are more efficient than the existing $Q_*$, and the difference is more significant for smaller $n$.

# 5 Concluding Remarks

We proposed two upper bounds $Q^n_{**}, Q^n_{***}$ and mathematically proved that they are tighter than $Q^n_*$ [7, 19]. As a result, in experiments that utilized the Alarm network, $Q^n_{**}, Q^n_{***}$ significantly outperformed $Q^n_*$ in

terms of efficiency. The former requires a slightly larger overhead than the latter, but the total time $T$ as well as the number $R$ of visited candidates for the former are much lower than those for the latter. In this sense, we can say that the overhead due to replacing $Q_*^n$ by $Q_{**}^n, Q_{***}^n$ is negligible compared with the gain from $R_*$ to $R_{**}, R_{***}$.

For the actual implementation, the largest array size is $2^{N-1}$. If we move those arrays from RAM to HDD, the largest array size will be $n$. In this sense, we may expect to execute BNSL in the high-dimensional setting $n \ll N$. In fact, the B&B procedure works better for smaller $n$, according to the results of this paper. However, in previous research, BNSL for large $N$ was investigated under general settings (not assuming anything about the sample size $n$). The tasks instigated by the current research to be undertaken in the (near) future involve realizing BNSL for large $N$ ($> 33$ [10], the current world record) and small $n$ (say, $n = 100$) because if $n$ is small, the B&B will work better and might save much more computational effort.

# References


[1] H. Akaike, "Information theory and an extension of the maximum likelihood principle, *2nd International Symposium on Information Theory*, Budapest, Hungary (1973).

[2] I. A. Beinlich, H. J. Suermondt, R. M. Chavez, and G. F. Cooper. "The ALARM Monitoring System: A Case Study with Two Probabilistic Inference Techniques for Belief Networks". In Proceedings of the 2nd European Conference on Artificial Intelligence in Medicine, pages 247-256. Springer-Verlag (1989).

[3] W. Buntine,"Theory refinement on Bayesian networks", *Uncertainty in Artificial Intelligence*, Los Angels, CA pages 52-60, (1991)

[4] J. Cussens and M. Bartlett, "GOBNILP 1.6.2 User/Developer Manual1", University of York, (2015).

[5] D. M. Chickering, C. Meek, and D. Heckerman. "Large-sample learning of Bayesian networks is NP-hard". *Uncertainty in Artificial Intelligence*, Acapulco, Mexico, pages 124-133 (2003).

[6] G. F. Cooper, E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data" it Machine Learning vol. 9, no. 4, pages 309-347 (1992).

[7] C. P. de Campos, Q. Ji, "Efficient Structure Learning of Bayesian Networks using Constraints", *Journal of Machine Learning Research*, vol. 12, no. Mar, pages 663?689, (2011).

[8] H. Jeffreys, *Theory of Probability*. Oxford University Press (1939).

[9] R.E. Krichevsky and V.K. Trofimov. 'The Performance of Universal Encoding", *IEEE Trans. Information Theory*, Vol. IT-27, No. 2, pp. 199-207 (1981)

[10] O. Nikolova, J. Zola, S. Aluru. "A Parallel Algorithm for Exact Structure Learning of Bayesian Networks". *Neural Information Processing Systems (NIPS)*, Workshop on Learning on Cores, Clusters and Clouds, 2010.

[11] J.Rissanen, "Modeling by shortest data description," *Automatica* 14: 465-471 (1978).

[12] T. Silander, P. Myllymaki, "A Simple Approach for Finding the Globally Optimal Bayesian Network Structure". *Uncertainty in Artificial Intelligence* (2006).

[13] A. P. Singh, A. W. Moore, "Finding optimal Bayesian networks by dynamic programming", Research Showcase@CMU (2005).

[14] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction and Search*. Springer Verlag, Berlin (1993).

[15] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference (Representation and Reasoning)*, Morgan Kaufmann Pub, 2nd edition (1988).

[16] J. Suzuki, "A Construction of Bayesian Networks from Databases on an MDL Principle". The Ninth Conference on Uncertainty in Artificial Intelligence, Washington D. C., 266-273 (1993).

[17] J. Suzuki, "Learning Bayesian Belief Networks Based on the Minimum Description Length Principle: An Efficient Algorithm Using the B & B Technique". in the proceedings of *International Conference on Machine Learning*, pages 462-470 (1996)

[18] J. Suzuki, "The Bayesian Chow-Liu Algorithm", in the proceedings of *The Sixth European Workshop on Probabilistic Graphical Models*, Granada, Spain (2012).

[19] J. Suzuki, "Efficiently Learning Bayesian Network Structures based on the B&B Strategy: A Theoretical Analysis", in the proceedings of *Advanced Methodologies for Bayesian Networks*, Yokohama, Japan (2015), published as Lecture Note on Artificial Intelligence 9095 (2016).

[20] J. Tian, "A branch-and-bound algorithm for MDL learning Bayesian networks". In the proceedings of *Uncertainty in Artificial Intelligence*, Palo Alto, pages 580-588 (2000).

[21] M. Ueno, "Robust learning Bayesian networks for prior belief", in the proceedings of *Uncertainty in Artificial Intelligence*, Corvallis, Oregon, pages: 698-707 (2011).