Finding All Bayesian Network Structures within a Factor of Optimal

Zhenyu A. Liao[1], Charupriya Sharma[1], James Cussens[2] and Peter van Beek[1]

[1]David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, ON Canada
{z6liao,c9sharma,vanbeek}@uwaterloo.ca

[2]Department of Computer Science
University of York
York, United Kingdom
james.cussens@york.ac.uk

Abstract

A Bayesian network is a widely used probabilistic graphical model with applications in knowledge discovery and prediction. Learning a Bayesian network (BN) from data can be cast as an optimization problem using the well-known score-and-search approach. However, selecting a single model (i.e., the best scoring BN) can be misleading or may not achieve the best possible accuracy. An alternative to committing to a single model is to perform some form of Bayesian or frequentist model averaging, where the space of possible BNs is sampled or enumerated in some fashion. Unfortunately, existing approaches for model averaging either severely restrict the structure of the Bayesian network or have only been shown to scale to networks with fewer than 30 random variables. In this paper, we propose a novel approach to model averaging inspired by performance guarantees in approximation algorithms. Our approach has two primary advantages. First, our approach only considers *credible* models in that they are optimal or near-optimal in score. Second, our approach is more efficient and scales to significantly larger Bayesian networks than existing approaches.

Introduction

A Bayesian network is a widely used probabilistic graphical model with applications in knowledge discovery, explanation, and prediction (**?**, **?**). A Bayesian network (BN) can be learned from data using the well-known *score-and-search* approach, where a scoring function is used to evaluate the fit of a proposed BN to the data, and the space of directed acyclic graphs (DAGs) is searched for the best-scoring BN. However, selecting a single model (i.e., the best-scoring BN) may not always be the best choice. When one is using BNs for knowledge discovery and explanation with limited data, selecting a single model may be misleading as there may be many other BNs that have scores that are very close to optimal and the posterior probability of even the best-scoring BN is often close to zero. As well, when one is using BNs for prediction, selecting a single model may not achieve the best possible accuracy.

An alternative to committing to a single model is to perform some form of Bayesian or frequentist model averaging (**?**, **?**, **?**). In the context of knowledge discovery, Bayesian model averaging allows one to estimate, for example, the posterior probability that an edge is present, rather than just knowing whether the edge is present in the best-scoring network. Previous work has proposed Bayesian and frequentist model averaging approaches to network structure learning that enumerate the space of all possible DAGs (**?**), sample from the space of all possible DAGs (**?**, **?**), consider the space of all DAGs consistent with a given ordering of the random variables (**?**, **?**), consider the space of tree-structured or other restricted DAGs (**?**, **?**), and consider only the k-best scoring DAGs for some given value of k (**?**, **?**, **?**, **?**, **?**, **?**). Unfortunately, these existing approaches either severely restrict the structure of the Bayesian network, such as only allowing tree-structured networks or only considering a single ordering, or have only been shown to scale to small Bayesian networks with fewer than 30 random variables.

In this paper, we propose a novel approach to model averaging for BN structure learning that is inspired by performance guarantees in approximation algorithms. Let OPT be the score of the optimal BN and assume without loss of generality that the optimization problem is to find the minimum-score BN. Instead of finding the k-best networks for some fixed value of k, we propose to find all Bayesian networks $\mathcal{G}$ that are within a factor ρ of optimal; i.e.,

$$OPT \leq score(\mathcal{G}) \leq \rho \cdot OPT, \qquad (1)$$

for some given value of $\rho \geq 1$, or equivalently,

$$OPT \leq score(\mathcal{G}) \leq OPT + \epsilon, \qquad (2)$$

for $\epsilon = (\rho - 1) \cdot OPT$. Instead of choosing arbitrary values for ϵ, $\epsilon \geq 0$, we show that for the two scoring functions BIC/MDL and BDeu, a good choice for the value of ϵ is closely related to the Bayes factor, a model selection criterion summarized in (**?**).

Our approach has two primary advantages. First, our approach only considers *credible* models in that they are optimal or near-optimal in score. Approaches that enumerate or sample from the space of all possible models consider DAGs with scores that can be far from optimal; for example, for the BIC/MDL scoring function the ratio of worst-scoring to best-scoring network can be four or five orders of magnitude[1]. A similar but more restricted case can be made

[1] Madigan and Raftery (**?**) deem such models *discredited* when they make a similar argument for not considering models whose probability is greater than a factor from the most probable.

against the approach which finds the k-best networks since there is no *a priori* way to know how to set the parameter k such that only credible networks are considered. Second, and perhaps most importantly, our approach is significantly more efficient and scales to Bayesian networks with almost 60 random variables. Existing methods for finding the optimal Bayesian network structure, e.g., (**?**, **?**) rely heavily for their success on a significant body of pruning rules that remove from consideration many candidate parent sets both before and during the search. We show that many of these pruning rules can be naturally generalized to preserve the Bayesian networks that are within a factor of optimal. We modify GOBNILP (**?**), a state-of-the-art method for finding an optimal Bayesian network, to implement our generalized pruning rules and to find all *near*-optimal networks. We show in an experimental evaluation that the modified GOBNILP scales to significantly larger networks without resorting to restricting the structure of the Bayesian networks that are learned.

Background

In this section, we briefly review the necessary background in Bayesian networks and scoring functions, and define the Bayesian network structure learning problem (for more background on these topics see (**?**, **?**)).

Bayesian Networks

A Bayesian network (BN) is a probabilistic graphical model that consists of a labeled directed acyclic graph (DAG), $G = (V, E)$ in which the vertices $V = \{V_1, \ldots, V_n\}$ correspond to n random variables, the edges E represent direct influence of one random variable on another, and each vertex V_i is labeled with a conditional probability distribution $P(V_i \mid \Pi_i)$ that specifies the dependence of the variable V_i on its set of parents Π_i in the DAG. A BN can alternatively be viewed as a factorized representation of the joint probability distribution over the random variables and as an encoding of the Markov condition on the nodes; i.e., given its parents, every variable is conditionally independent of its non-descendents.

Each random variable V_i has state space $\Omega_i = \{v_{i1}, \ldots, v_{ir_i}\}$, where r_i is the cardinality of Ω_i and typically $r_i \geq 2$. Each Π_i has state space $\Omega_{\Pi_i} = \{\pi_{i1}, \ldots, \pi_{ir_{\Pi_i}}\}$. We use r_{Π_i} to refer to the number of possible instantiations of the parent set Π_i of V_i (see Figure 1). The set $\theta = \{\theta_{ijk}\}$ for all $i = \{1, \ldots, n\}, j = \{1, \ldots, r_{\Pi_i}\}$ and $k = \{1, \ldots, r_i\}$ represents parameters in G where each element in θ, $\theta_{ijk} = P(v_{ik} \mid \pi_{ij})$.

The predominant method for Bayesian network structure learning (BNSL) from data is the *score-and-search* method. Let $I = \{I_1, \ldots, I_N\}$ be a dataset where each instance I_i is an n-tuple that is a complete instantiation of the variables in V. A *scoring function* $\sigma(G \mid I)$ assigns a real value measuring the quality of $G = (V,E)$ given the data I. Without loss of generality, we assume that a lower score represents a better quality network structure and omit I when the data is clear from context.

Definition 1. *Given a non-negative constant ϵ and a dataset $I = \{I_1, \ldots, I_N\}$, a* ***credible network*** *G is a network that has a score $\sigma(G)$ such that $OPT \leq \sigma(G) \leq OPT + \epsilon$, where OPT is the score of the optimal Bayesian network.*

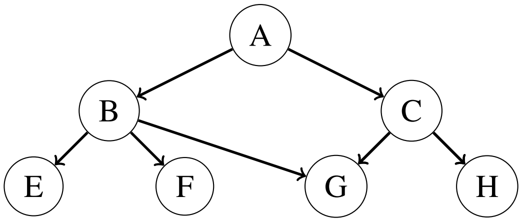

Figure 1: Example Bayesian network: Variables A, B, F and G have the state space $\{0, 1\}$. The variables C and E have state space $\{0, 1, 3\}$ and H has state space $\{2, 4\}$ Thus $r_A = r_B = r_F = r_G = 2$, $r_C = r_E = 3$ and $r_H = 2$. Consider the parent set of G, $\Pi_G = \{B, C\}$ The state space of Π_G is $\Omega_{\Pi_G} = \{\{0, 0\}, \{0, 1\}, \{0, 3\}, \{1, 0\}, \{1, 1\}, \{1, 3\}\}$. and $r_{\Pi_G} = 6$.

In this paper, we focus on solving a problem we call the ϵ-Bayesian Network Structure Learning (ϵBNSL). Note that the BNSL for the optimal network(s) is a special case of ϵBNSL where $\epsilon = 0$.

Definition 2. *Given a non-negative constant ϵ, a dataset $I = \{I_1, \ldots, I_N\}$ over random variables $V = \{V_1, \ldots, V_n\}$ and a scoring function σ, the ϵ-Bayesian Network Structure Learning (ϵBNSL) problem is to find all credible networks.*

Scoring Functions

Scoring functions usually balance goodness of fit to the data with a penalty term for model complexity to avoid overfitting. Common scoring functions include BIC/MDL (**?**, **?**) and BDeu (**?**, **?**). An important property of these (and most) scoring functions is decomposability, where the score of the entire network $\sigma(G)$ can be rewritten as the sum of local scores associated to each vertex $\sum_{i=1}^{n} \sigma(V_i, \Pi_i)$ that only depends on V_i and its parent set Π_i in G. The local score is abbreviated below as $\sigma(\Pi_i)$ when the local node V_i is clear from context. Pruning techniques can be used to reduce the number of candidate parent sets that need to be considered, but in the worst-case the number of candidate parent sets for each variable V_i is exponential in n, where n is the number of vertices in the DAG.

In this work, we focus on the Bayesian Information Criterion (BIC) and the Bayesian Dirichlet, specifically BDeu, scoring functions. The BIC scoring function in this paper is defined as,

$$BIC : \sigma(G) = \max_{\theta} L_{G,I}(\theta) - t(G) \cdot w.$$

Here, $w = \frac{\log N}{2}$, $t(G)$ is a penalty term and $L_{G,I}(\theta)$ is the log likelihood, given by,

$$L_{G,I}(\theta) = \sum_{i=1}^{n} \sum_{j=1}^{r_{\Pi_i}} \sum_{k=1}^{r_i} \log \theta_{ijk}^{n_{ijk}},$$

where n_{ijk} is the number of instances in I where v_{ik} and π_{ij} co-occur. As the BIC function is decomposable, we can associate a score to Π_i, a candidate parent set of V_i as follows,

$$BIC : \sigma(\Pi_i) = \max_{\theta_i} L(\theta_i) - t(\Pi_i) \cdot w.$$

Here, $L(\theta_i) = \sum_{j=1}^{r_{\Pi_i}} \sum_{k=1}^{r_i} n_{ijk} \log \theta_{ijk}$ and $t(\Pi_i) = r_{\Pi_i}(r_i - 1)$. The BDeu scoring function in this paper is defined as,

$$\begin{aligned} BDeu : \sigma(G) = &\sum_{i=1}^{n} \sum_{j=1}^{r_{\Pi_i}} \log \frac{\Gamma(\alpha)}{\Gamma(\alpha + n_{ij})} \\ &+ \sum_{i=1}^{n} \sum_{j=1}^{r_{\Pi_i}} \sum_{k=1}^{r_i} \log \frac{\Gamma(\frac{\alpha}{r_i} + n_{ijk})}{\Gamma(\frac{\alpha}{r_i})}, \end{aligned}$$

where α is the equivalent sample size and $n_{ij} = \sum_k n_{ijk}$. As the BDeu function is decomposable, we can associate a score to Π_i, a candidate parent set of V_i as follows,

$$\begin{aligned} BDeu : \sigma(\Pi_i) = \sum_{j=1}^{r_{\Pi_i}} \Bigg(&\log \frac{\Gamma(\alpha)}{\Gamma(\alpha + n_{ij})} \\ &+ \sum_{k=1}^{r_i} \log \frac{\Gamma(\frac{\alpha}{r_i} + n_{ijk})}{\Gamma(\frac{\alpha}{r_i})} \Bigg). \end{aligned}$$

The Bayes Factor

In this section, we show that a good choice for the value of ϵ for the ϵBNSL problem is closely related to the Bayes factor (BF), a model selection criterion summarized in (Kass and Raftery 1995).

The BF was proposed by Jeffreys as an alternative to significance test (**?**). It was thoroughly examined as a practical model selection tool in (**?**). Let G_0 and G_1 be DAGs (BNs) in the set of all DAGs $\mathcal{G}$ defined over V. The BF in the context of BNs is defined as,

$$BF(G_0, G_1) = \frac{P(I \mid G_0)}{P(I \mid G_1)},$$

namely the odds of the probability of the data predicted by network G_0 and G_1. The actual calculation of the BF often relies on Bayes' Theorem as follows,

$$\frac{P(G_0 \mid I)}{P(G_1 \mid I)} = \frac{P(I \mid G_0)}{P(I \mid G_1)} \cdot \frac{P(G_0)}{P(G_1)} = \frac{P(I, G_0)}{P(I, G_1)}.$$

Since it is typical to assume the prior over models is uniform in ϵBNSL, the BF can then be obtained using either $P(G \mid I)$ or $P(I, G) \forall G \in \mathcal{G}$. We use those two representations to show how BIC and BDeu scores relate to the BF.

Using Laplace approximation and other simplifications in (**?**), Ripley derived the following approximation to the logarithm of the marginal likelihood for network G (a similar derivation is given in (**?**)),

$$\begin{aligned} \log P(I \mid G) = &L_{G,I}(\hat{\theta}) - t(G) \cdot \frac{\log N}{2} + t(G) \cdot \frac{\log 2\pi}{2} \\ &- \frac{1}{2} \log |J_{G,I}(\hat{\theta})| + \log P(\hat{\theta} \mid G), \end{aligned}$$

where $\hat{\theta}$ is the maximum likelihood estimate of model parameters and $J_{G,I}(\hat{\theta})$ is the Hessian matrix evaluated at $\hat{\theta}$. It follows that,

$$\log P(I \mid G) = -BIC(I, G) + O(1).$$

The above equation shows that the BIC score was designed to approximate the log marginal likelihood. If we drop the lower-order term, we can then obtain the following equation,

$$\begin{aligned} BIC(I, G_1) - BIC(I, G_0) &= \log \frac{P(I \mid G_0)}{P(I \mid G_1)} \\ &= \log BF(G_0, G_1). \end{aligned}$$

It has been indicated in (**?**) that as $N \to \infty$, the difference of the two BIC scores, dubbed the Schwarz criterion, approaches the true value of $\log BF$ such that,

$$\frac{BIC(I, G_1) - BIC(I, G_0) - \log BF(G_0, G_1)}{\log BF(G_0, G_1)} \to 0.$$

Therefore, the difference of two BIC scores can be used as a rough approximation to $\log BF$. Note that some papers define BIC to be twice as large as the BIC defined in this paper, but the above relationship still holds albeit with twice the logarithm of the BF.

Similarly, the difference of the BDeu scores can be expressed in terms of the BF. In fact, the BDeu score is the log marginal likelihood where there are Dirichlet distributions over the parameters (**?**, **?**); i.e.,

$$\log P(I, G) = -BDeu(I, G),$$

and thus,

$$\begin{aligned} BDeu(I, G_1) - BDeu(I, G_0) &= \log \frac{P(I, G_0)}{P(I, G_1)} \\ &= \log BF(G_0, G_1). \end{aligned}$$

The above results are consistent with the observation in (**?**) that the $\log BF$ can be interpreted as a measure for the *relative success* of two models at predicting data, sometimes referred to as the "weight of evidence", without assuming either model is true. The desired value of BF, however, is often specific to a study and determined with domain knowledge, e.g., a BF of 1000 is more appropriate in forensic science. **?** (**?**) proposed the following interpreting scale for the BF: a BF of 1 to 3 bears only anecdotal evidence, a BF of 3 to 20 suggests some positive evidence that G_0 is better, a BF of 20 to 150 suggests strong evidence in favor of G_0, and a BF greater than 150 indicates very strong evidence. If we deem 20 to be the desired BF in ϵBNSL, i.e., $G_0 = G^*$ and $\epsilon = \log(20)$, then any network with a score less than $\log(20)$ away from the optimal score would be *credible*, otherwise it would be *discredited*. Note that the ratio of posterior probabilities was defined as λ in (**?**, **?**) and was used as a metric to assess arbitrary values of k in finding the k-best networks.

Finally, the ϵBNSL problem using the BIC or BDeu scoring function given a desired BF can be written as,

$$OPT \leq \mathit{score}(\mathcal{G}) \leq OPT + \log BF. \tag{3}$$

Pruning Rules for Candidate Parent Sets

To find all near-optimal BNs given a BF, the local score $\sigma(\Pi_i)$ for each candidate parent set $\Pi_i \subseteq 2^{V-\{V_i\}}$ and each random variable V_i must be computed. As this is very cost prohibitive, the search space of candidate parent sets can be pruned, provided that global optimality constraints are not violated.

A candidate parent set Π_i can be *safely pruned* given a non-negative constant $\epsilon \in \mathbb{R}^+$ if Π_i cannot be the parent set of V_i in any network in the set of credible networks. Note that for $\epsilon = 0$, the set of credible networks just contains the optimal network(s). We discuss the original rules and their generalization below and proofs for them can be found in the *supplemental material*.

? (**?**) gave a pruning rule for all decomposable scoring functions. This rule compares the score of a candidate parent set to those of its subsets. We give a relaxed version of the rule.

Lemma 1. *Given a vertex variable V_j, candidate parent sets Π_j and Π'_j, and some $\epsilon \in \mathbb{R}^+$, if $\Pi_j \subset \Pi'_j$ and $\sigma(\Pi_j) + \epsilon \geq \sigma(\Pi'_j)$, Π'_j can be safely pruned.*

Pruning with BIC/MDL Score

A pruning rule comparing the BIC score and penalty associated to a candidate parent set to those of its subsets was introduced in (**?**). The following theorem gives a relaxed version of that rule.

Theorem 1. *Given a vertex variable V_j, candidate parent sets Π_j and Π'_j, and some $\epsilon \in \mathbb{R}^+$, if $\Pi_j \subset \Pi'_j$ and $\sigma(\Pi_j) - t(\Pi'_j) + \epsilon < 0$, Π'_j and all supersets of Π'_j can be safely pruned if σ is the BIC function.*

Another pruning rule for BIC appears in (**?**). This provides a bound on the number of possible instantiations of subsets of a candidate parent set. The following theorem relaxes that rule.

Theorem 2. *Given a vertex variable V_i, and a candidate parent set Π_i such that $r_{\Pi_i} > \frac{N}{w}\frac{\log r_i}{r_i - 1} + \epsilon$ for some $\epsilon \in \mathbb{R}^+$, if $\Pi_i \subsetneq \Pi'_i$, then Π'_i can be safely pruned if σ is the BIC scoring function.*

The following corollary of Theorem 2 gives a useful upper bound on the size of a candidate parent set.

Corollary 1. *Given a vertex variable V_i and candidate parent set Π_i, if Π_i has more than $\lceil \log_2 N + \epsilon \rceil$ elements, for some $\epsilon \in \mathbb{R}^+$, Π_i can be safely pruned if σ is the BIC scoring function.*

Corollary 1 provides an upper-bound on the size of parent sets based solely on the sample size. The following table summarizes such an upper-bound given different amounts of data N and a BF of 20.

N	100	500	10^3	5×10^3	10^4	5×10^4	10^5
$\lvert\Pi\rvert$	10	12	13	16	17	19	20

The entropy of a candidate parent set is also a useful measure for pruning. A pruning rule, given by (**?**), provides an upper bound on conditional entropy of candidate parent sets and their subsets. We give a relaxed version of their rule. First, we note that entropy for a vertex variable V_i is given by,

$$H(V_i) = -\sum_{k=1}^{r_i} \frac{n_{ik}}{N} \log \frac{n_{ik}}{N},$$

where n_{ik} represents how many instances in the dataset contain v_{ik}, where v_{ik} is an element in the state space Ω_i of V_i. Similarly, entropy for a candidate parent set Π_i is given by,

$$H(\Pi_i) = -\sum_{j=1}^{r_{\Pi_i}} \frac{n_{ij}}{N} \log \frac{n_{ij}}{N}.$$

Conditional information is given by,

$$H(X \mid Y) = H(X \cup Y) - H(Y).$$

Theorem 3. *Given a vertex variable V_i, and candidate parent set Π_i, let $V_j \notin \Pi_i$ such that $N \cdot \min\{H(V_i \mid \Pi_i), H(V_j \mid \Pi_i)\} \geq (1 - r_j) \cdot t(\Pi_i) + \epsilon$ for some $\epsilon \in \mathbb{R}^+$. Then the candidate parent set $\Pi'_i = \Pi_i \cup \{V_j\}$ and all its supersets can be safely pruned if σ is the BIC scoring function.*

Pruning with BDeu Score

A pruning rule for the BDeu scoring function appears in (**?**) and a more general version is included in (**?**). Here, we present a relaxed version of the rule in (**?**).

Theorem 4. *Given a vertex variable V_i and candidate parent sets Π_i and Π'_i such that $\Pi_i \subset \Pi'_i$ and $\Pi_i \neq \Pi'_i$, let $r_i^+(\Pi'_i)$ be the number of positive counts in the contingency table for Π'_i. If $\sigma(\Pi_i) + \epsilon < r_i^+(\Pi'_i) \log r_i$, for some $\epsilon \in \mathbb{R}^+$ then Π'_i and the supersets of Π'_i can be safely pruned if σ is the BDeu scoring function..*

Experimental Evaluation

In this section, we evaluate the proposed BF based method and compare its performance with published k-best solvers.

Our proposed method is more memory efficient comparing to the k-best based solvers in BDeu scoring and often collects more networks in a shorter period of time. With the pruning rules generalized above, our method can scale up to datasets with 57 variables in BIC scoring, whereas the previous best results are reported on a network of 29 variables using the k-best approach with score pruning (**?**).

The datasets are obtained from the UCI Machine Learning Repository (**?**) and the Bayesian Network Repository[2]. Some of the complete local scoring files are downloaded from the GOBNILP website[3] and are used for the k-best related experiments only. Since not all solvers in the k-best experiments can take in scoring files, we exclude the time to compute local scores from the comparison. Both BIC/MDL (**?**, **?**) and BDeu (**?**, **?**) scoring functions are used where applicable. All experiments are conducted on computers with 2.2 GHz Intel E7-4850V3 processors. Each experiment is limited to 64 GB of memory and 24 hours of CPU time.

[2] `http://www.bnlearn.com/bnrepository/`

[3] `https://www.cs.york.ac.uk/aig/sw/gobnilp/#benchmarks`

Data	n	N	T_3 (s)	$\|\mathcal{G}_3\|$	$\|\mathcal{M}_3\|$	T_{20} (s)	$\|\mathcal{G}_{20}\|$	$\|\mathcal{M}_{20}\|$	T_{150} (s)	$\|\mathcal{G}_{150}\|$	$\|\mathcal{M}_{150}\|$
tic tac toe	10	958	1.9	192	64	2.0	192	64	3.3	544	160
wine	14	178	4.1	308	51	24.9	3,449	576	143.7	26,197	4,497
adult	14	32,561	17.5	324	162	45.1	1,140	570	55.7	2,281	1,137
nltcs	16	3,236	53.8	240	120	201.7	1,200	600	1,005.1	4,606	2,303
msnbc	17	58,265	3,483.0	24	24	7,146.9	960	504	8,821.4	1,938	1,026
letter	17	20,000	OT	—	—	OT	—	—	OT	—	—
voting	17	435	1.3	27	2	4.0	441	33	14.3	2,222	170
zoo	17	101	8.1	49	13	21.9	1,111	270	299.3	21,683	5,392
hepatitis	20	155	7.1	580	105	513.3	87,169	15,358	1,452.8	150,000	49,269
parkinsons	23	195	30.7	1,088	336	3,165.9	150,000	39,720	4,534.3	150,000	116,206
sensors	25	5456	OT	—	—	OT	—	—	OT	—	—
autos	26	159	95.0	560	200	2,382.8	50,374	17,790	6,666.9	150,000	54,579
insurance	27	1,000	49.8	8,226	2,062	244.9	104,870	25,580	414.5	148,925	36,072
horse	28	300	18.8	1,643	246	1,358.8	150,000	28,186	1,962.5	150,000	69,309
flag	29	194	16.1	773	169	4,051.9	150,000	39,428	5,560.9	150,000	122,185
wdbc	31	569	396.1	398	107	10,144.2	28,424	8,182	45,938.2	150,000	54,846
mildew	35	1000	1.2	1,026	2	1.2	1,026	2	2.1	2,052	4
soybean	36	266	7,729.4	150,000	150,000	16,096.8	150,000	62,704	8,893.5	150,000	118,368
alarm	37	1000	6.3	1,508	122	684.2	123,352	9,323	2,258.4	150,000	8,484
bands	39	277	100.9	7,092	810	2,032.6	150,000	44,899	16,974.8	150,000	95,774
spectf	45	267	432.4	27,770	4,510	7,425.2	150,000	51,871	19,664.8	150,000	63,965
sponge	45	76	16.8	1,102	65	1,301.0	146,097	7,905	1,254.4	150,000	90,005
barley	48	1000	0.8	182	1	0.8	364	2	1.3	1,274	5
hailfinder	56	100	171.5	150,000	20	149.4	150,000	748	214.6	150,000	294
hailfinder	56	500	286.1	150,000	30,720	314.1	150,000	18,432	217.3	150,000	24,576
lung cancer	57	32	584.3	150,000	40,621	966.6	150,000	79,680	2,739.7	150,000	48,236

Table 1: The search time T, the number of collected networks $|\mathcal{G}|$ and the number of MECs $|\mathcal{M}|$ in the collected networks at BF = 3, 20 and 150 using BIC, where n is the number of random variables in the dataset, N is the number of instances in the dataset and OT = Out of Time.

The Bayes Factor Approach

We modified the development version (9c9f3e6) of GOBNILP, referred below as GOBNILP_dev, to apply pruning rules presented above during scoring and supplied appropriate parameter settings for collecting near-optimal networks[4]. The code is compiled with SCIP 6.0.0 and CPLEX 12.8.0. GOBNILP extends the SCIP Optimization Suite (**?**) by adding a *constraint handler* for handling the acyclicity constraint for DAGs. If multiple BNs are required GOBNILP_dev just calls SCIP to ask it to collect feasible solutions. In this mode, when SCIP finds a solution, the solution is stored, a constraint is added to render that solution infeasible and the search continues. This differs from (and is much more efficient than) GOBNILP's current method for finding k-best BNs where an entirely new search is started each time a new BN is found. A recent version of SCIP has a separate "reoptimization" method which might allow better k-best performance for GOBNILP but we do not explore that here. By default when SCIP is asked to collect solutions it turns off all cutting plane algorithms. This led to very poor GOBNILP performance since GOBNILP relies on cutting plane generation. Therefore, this default setting is overridden in GOBNILP_dev to allow cutting planes when collecting solutions. To find only solutions with objective no worse than $(OPT + \epsilon)$, SCIP's `SCIPsetObjlimit` function is used. Note that, for efficiency reasons, this is **not** effected by adding a linear constraint.

We first use GOBNILP_dev to find the optimal scores since GOBNILP_dev takes objective limit $(OPT + \epsilon)$ for enumerating feasible networks. Then all networks falling into the limit are collected with a counting limit of 150,000. Finally the collected networks are categorized into Markov equivalence classes (MECs), where two networks belong to the same MEC iff they have the same skeleton and v-structures (**?**). The proposed approach is tested on datasets with up to 57 variables. The search time T, the number of collected networks $|\mathcal{G}|$ and the number of MECs $\mathcal{M}$ in the collected networks at BF = 3, 20 and 150 using BIC are reported in Table 1, where n is the number of random variables in the dataset and N is the number of instances in the dataset. The three thresholds are chosen according to the interpreting scale suggested by (**?**) where 3 marks the difference between anecdotal and positive evidence, 20 marks positive and strong evidence and 150 marks strong and very strong evidence. The search time mostly depends on a combined effect of the size of the network, the sample size and the number of MECs at a given BF. Some fairly large networks such as alarm, sponge and barley are solved much faster than smaller networks with a large sample size, e.g., msnbc and letter.

The results also indicate that the number of collected networks and the number of MECs at three BF levels varies substantially across different datasets. In general, datasets with smaller sample sizes tend to have more networks collected at a given BF since near-optimal networks have simi-

[4] The modified code is available at: `https://www.cs.york.ac.uk/aig/sw/gobnilp/`

Data	n	N	T_k (s)	k	T_{EC} (s)	$\lvert\mathcal{G}_k\rvert$	T_{20} (s)	$\lvert\mathcal{G}_{20}\rvert$	$\lvert\mathcal{M}_{20}\rvert$
tic tac toe	10	958	0.2	10	0.5	67	0.6	152	24
			2.8	100	6.0	673			
			70.7	1,000	78.5	7,604			
wine	14	178	3.4	10	12.0	60	35.9	8,734	6,262
			85.0	100	168.4	448			
			3,420.4	1,000	3,064.4	4,142			
adult	14	32,561	3.3	10	633.5	68	9.3	792	19
			73.6	100	63,328.9	1,340			
			2,122.8	1,000	OT	—			
nltcs	16	3,236	11.8	10	47,338.4	552	125.5	652	326
			406.6	100	OT	—			
			13,224.6	1,000	OT	—			
msnbc	17	58,265	ES	—	ES	—	4,018.9	24	24
letter	17	20,000	26.0	10	18,788.0	200	56,344.8	20	10
			909.8	100	OT	—			
			41,503.9	1,000	OT	—			
voting	17	435	34.1	10	101.9	30	6.0	621	207
			1,125.7	100	1,829.2	3,392			
			38,516.2	1,000	42,415.3	3,665			
zoo	17	101	33.5	10	99.8	52	8,418.8	29,073	6,761
			1,041.7	100	1,843.4	100			
			41,412.1	1,000	OT	—			
hepatitis	20	155	351.2	10	872.3	89	441.4	28,024	3,534
			13,560.3	100	20,244.7	842			
			OT	1,000	OT	—			
parkinsons	23	195	3,908.2	10	OT	—	1,515.9	150,000	42,448
			OT	100	OT	—			
			OT	1,000	OT	—			
autos	26	159	OM	1	OM	—	OT	—	—
insurance	27	1,000	OM	1	OM	—	8.3	1,081	133

Table 2: The search time T and the number of collected networks k, $|\mathcal{G}_k|$ and $|\mathcal{G}_{20}|$ for KBest, KbestEC and GOBNILP_dev (BF = 20) using BDeu, where n is the number of random variables in the dataset, N is the number of instances in the dataset, OM = Out of Memory, OT = Out of Time and ES = Error in Scoring. Note that $|\mathcal{G}_k|$ is the number of DAGs covered by the k-best MECs in KBestEC and $|\mathcal{M}_{20}|$ is the number of MECs in the networks collected by GOBNILP_dev.

lar posterior probabilities to the best network. Although the desired level of BF for a study, like the p-value, is often determined with domain knowledge, the proposed approach, given sufficient samples, will produce meaningful results that can be used for further analysis.

Bayes Factor vs. k-Best

In this section, we compare our approach with published solvers that are able to find a subset of top-scoring networks with the given parameter k. The solvers under consideration are KBest_12b[5] from (?), KBestEC[6] from (?), and GOBNILP 1.6.3 (?), referred to as KBest, KBestEC and GOBNILP below. The first two solvers are based on the dynamic programming approach introduced in (?). Due to the lack of support for BIC in KBest and KBestEC, only BDeu with a equivalent sample size of one is used in corresponding experiments.

The most recent stable version of GOBNILP is 1.6.3 that works with SCIP 3.2.1. The default configuration is used and experiments are conducted for both BIC and BDeu scoring functions. However, the k-best results are omitted here due to its poor performance. Despite that GOBNILP can iteratively find the k-best networks in descending order by adding linear constraints, the pruning rules designed to find the best network are turned off to preserve sub-optimal networks. In fact, the memory usage often exceeded 64 GB during the initial ILP formulation, indicating that the lack of pruning rules posed serious challenge for GOBNILP. GOBNILP_dev, on the other hand, can take advantage of the pruning rules presented above in the proposed BF approach and its results compare favorably to KBest and KBestEC.

The experimental results of KBest, KBestEC and GOBNILP_dev are reported in Table 2, where n is the number of random variables in the dataset, N is the number of instances in the dataset, and k is the number of top scoring networks. The search time T is reported for KBest, KBestEC and GOBNILP_dev (BF = 20). The number of DAGs covered by the k MECs $|\mathcal{G}_k|$ is reported for KBestEC. In comparison, the last two columns are the number of found networks $|\mathcal{G}_{20}|$ and the number of MECs $|\mathcal{M}_{20}|$ using the BF approach with a given BF of 20 and BDeu scoring function.

As the number of requested networks k increases, the

[5] http://web.cs.iastate.edu/~jtian/Software/UAI-10/KBest.htm

[6] http://web.cs.iastate.edu/~jtian/Software/AAAI-14-yetian/KBestEC.htm

search time for both KBest and KBestEC grows exponentially. The KBest and KBestEC are designed to solve problems of size fewer than 20[7], and so they have some difficulty with larger datasets. They also fail to generate correct scoring files for msnbc. KBestEC seems to successfully expand the coverage of DAGs with some overhead for checking equivalence classes. However, KBestEC took much longer than KBest for some instances, e.g., nltcs and letter, and the number of DAGs covered by the found MECs is inconsistent for nltcs, letter and zoo. The search time for the BF approach is improved over the k-best approach except for datasets with very large sample sizes. The generalized pruning rules are very effective in reducing the search space, which then allows GOBNILP_dev to solve the ILP problem subsequently. Comparing to the improved results in (**?**, **?**; **?**), our approach can scale to larger networks if the scoring file can be generated.[8]

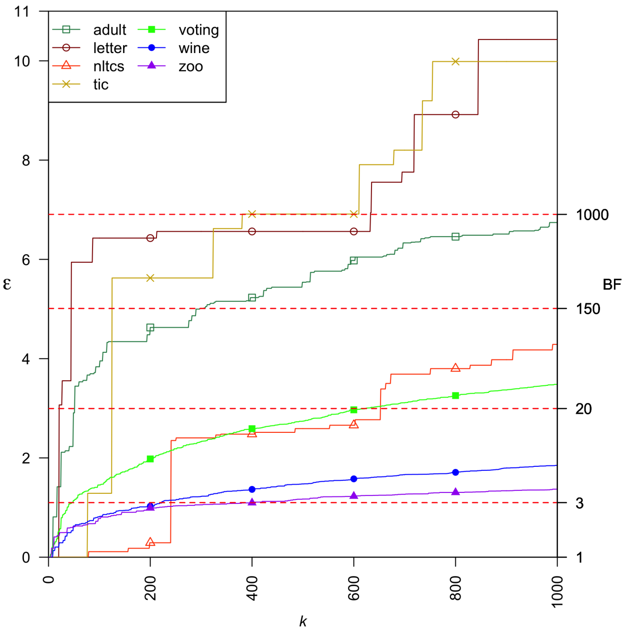

Figure 2: The deviation ϵ from the optimal BDeu score by k using results from KBest. The corresponding values of the BF ($\epsilon = \log(BF)$, see Equation 3) are presented on the right. For example, if the desired BF value is 20, then all networks falling below the dash line at 20 are credible.

Now we show that different datasets have distinct score patterns in the top scoring networks. The scores of the 1,000-best networks for some datasets in the KBest experiment are plotted in Figure 2. A specific line for a dataset indicates the deviation ϵ from the optimal BDeu score by the kth-best network. For reference, the red dash lines represent different levels of BFs calculated by $\epsilon = \log BF$ (See Equation 3). The figure shows that it is difficult to pick a value for k *a priori* to capture the appropriate set of top scoring networks. For a few datasets such as adult and letter, it only takes fewer than 50 networks to reach a BF of 20, whereas zoo needs more than 10,000 networks. The sample size has a significant effect on the number of networks at a given BF since the lack of data leads to many BNs with similar probabilities. It would be reasonable to choose a large value for k in model averaging when data is scarce and vice versa, but only the BF approach is able to automatically find the appropriate and credible set of networks for further analysis.

Conclusion

Existing approaches for model averaging for Bayesian network structure learning either severely restrict the structure of the Bayesian network or have only been shown to scale to networks with fewer than 30 random variables. In this paper, we proposed a novel approach to model averaging inspired by performance guarantees in approximation algorithms that considers all networks within a factor of optimal. Our approach has two primary advantages. First, our approach only considers *credible* models in that they are optimal or near-optimal in score. Second, our approach is significantly more efficient and scales to much larger Bayesian networks than existing approaches. We modified GOBNILP (**?**), a state-of-the-art method for finding an optimal Bayesian network, to implement our generalized pruning rules and to find all *near*-optimal networks. Our experimental results demonstrate that the modified GOBNILP scales to significantly larger networks without resorting to restricting the structure of the Bayesian networks that are learned.

Sed impedit blanditiis exercitationem natus voluptates at vel eos, ut vitae dolorum odit molestiae iusto.Officiis doloremque quibusdam enim dolorem eos sequi autem deserunt, cupiditate nihil id dignissimos unde accusantium soluta labore, incidunt eaque dolorum aperiam molestiae.Totam reprehenderit a dicta at deleniti qui, maiores aperiam necessitatibus nesciunt sit expedita quia quidem, suscipit molestiae fugit consectetur, deserunt molestiae nobis ex molestias ratione quaerat cumque ipsam, et eveniet veritatis officia aliquid dolorum eaque enim amet.Iusto incidunt facilis vitae molestias ex quasi illo numquam, distinctio ut iste id commodi fugiat explicabo numquam accusamus non earum repellat, blanditiis sed quam repudiandae totam laboriosam temporibus a maxime pariatur, dolorum magni deleniti earum facere nemo perspiciatis, nulla illo dicta fugiat culpa ab iure sunt animi harum possimus nostrum.Maxime eos autem commodi ratione odit perspiciatis iste, in voluptatum hic corrupti minima.Nam nisi dolorum provident dicta reiciendis nulla, voluptas voluptatibus sint deleniti ratione quibusdam aut iste corrupti, deleniti doloribus quas molestiae pariatur maxime cum ducimus ea delectus tenetur?Earum commodi alias repudiandae laborum quod officia provident dolorem, magnam voluptatibus repellat quos facilis blanditiis ad ipsa modi explicabo culpa porro, enim aperiam modi aliquid, quo doloribus corrupti illo id praesentium, dolorum expedita ratione ducimus fugit?Sed iure nam quibusdam officiis et aliquid, at ab asperiores possimus ad consequatur itaque architecto, error quo dignissimos commodi alias optio laudantium quod est

[7] Obtained through correspondence with the author.

[8] We are unable to generate BDeu score files for datasets with over 30 variables.

iusto.Quos nam reiciendis asperiores doloremque molestiae
cum perspiciatis tempora nemo, nisi illo aut