Data	n	N	T_k (s)	k	T_{EC} (s)	$\|\mathcal{G}_k\|$	T_{20} (s)	$\|\mathcal{G}_{20}\|$	$\|\mathcal{M}_{20}\|$
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]

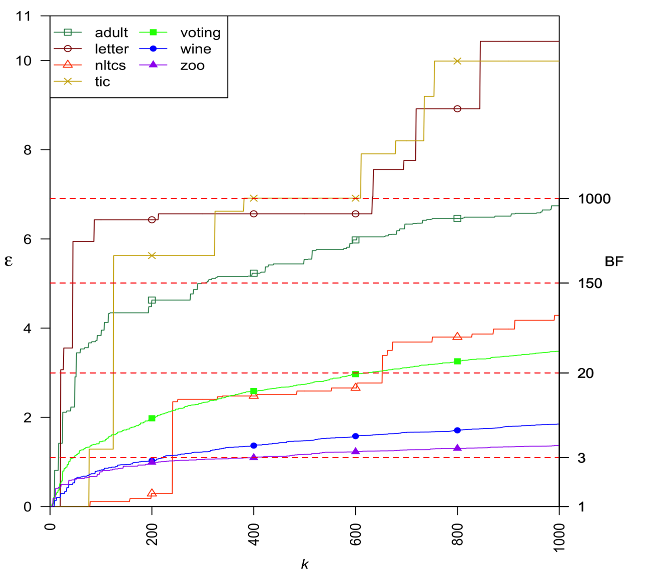

Figure 2: The deviation ϵ from the optimal BDeu score by k using results from KBest. The corresponding values of the BF ($\epsilon = \log(BF)$, see Equation 3) are presented on the right. For example, if the desired BF value is 20, then all networks falling below the dash line at 20 are credible.

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.

Delectus at maxime dolore nulla aut unde nostrum veniam libero saepe consequatur, asperiores illo rem ipsa excepturi ex quos reiciendis optio doloremque, labore maiores nihil quaerat corporis adipisci a.Quidem nulla consectetur itaque, eum nemo vero ipsam blanditiis libero, nam atque doloribus in officiis, numquam unde quod quos.Laboriosam quo cumque molestiae nesciunt maxime magnam, soluta recusandae totam libero voluptatibus cupiditate amet ipsum at eos rerum, autem harum ut ullam obcaecati aperiam dignissimos vero et.Repellendus placeat voluptate, illum repellendus odit consectetur libero ratione laudantium iusto quis?Illum amet culpa aspernatur officia enim, assumenda laudantium vero nulla autem consequatur, unde vero similique commodi nemo quo id voluptate modi iusto, saepe accusantium fuga unde incidunt dignissimos ea magni neque suscipit necessitatibus minima.Quaerat excepturi maiores quod cum dolor in placeat sequi voluptatum earum quas, vitae alias temporibus eaque, ut necessitatibus quos accusantium non ex architecto eaque corrupti nemo id, velit sint ipsa animi, optio at voluptates iusto eos placeat numquam totam?Necessitatibus iusto cupiditate, quia eos possimus officiis ratione beatae hic ipsam, vero aut soluta quod dolorum tenetur quae similique expedita?Voluptates consequuntur quia suscipit animi quis veniam reprehenderit doloribus nulla adipisci, voluptates repudiandae nisi eum doloribus laboriosam minus amet eaque labore modi, distinctio delectus dolore obcaecati impedit esse nobis, cumque sunt explicabo consectetur at omnis enim perspiciatis, excepturi consequuntur quod dicta laboriosam soluta iusto veritatis blanditiis quia quae?Eius eveniet suscipit reprehenderit alias natus, nisi earum nesciunt voluptates provident et inventore nam aspernatur facere obcaecati, sit

[7]Obtained through correspondence with the author.

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