International Journal of Approximate Reasoning 92 (2018) 363–375

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

On pruning with the MDL Score ☆

Eunice Yuh-Jie Chen *, Adnan Darwiche, Arthur Choi

Computer Science Department, University of California, Los Angeles, United States

ARTICLE INFO

Article history:
Received 10 January 2017
Received in revised form 27 August 2017
Accepted 13 October 2017
Available online 27 October 2017

Keywords:
Bayesian networks
Structure learning

ABSTRACT

The space of Bayesian network structures is prohibitively large and hence numerous techniques have been developed to prune this search space, but without eliminating the optimal structure. Such techniques are critical for structure learning to scale to larger datasets with more variables. Prior works exploited properties of the MDL score to prune away large regions of the search space that can be safely ignored by optimal structure learning algorithms. In this paper, we propose new techniques for pruning regions of the search space that can be safely ignored by algorithms that enumerate the k-best Bayesian network structures. Empirically, these techniques allow a state-of-the-art structure enumeration algorithm to scale to datasets with significantly more variables.

1. Introduction

Learning the structure of a Bayesian network is a fundamental problem in artificial intelligence and machine learning. In particular, we seek a structure, a directed acyclic graph (DAG), that best explains a given dataset [9,14,15]. In practice, learning a single optimal DAG may not be sufficient, especially when the dataset has few examples and is otherwise noisy. Thus, we are interested in discovering other likely DAGs, and not just the best one.

Recently, a number of algorithms have been proposed to *enumerate* the k-most likely DAGs from a given dataset [22,8,4,2,3]. For example, using dynamic programming, we can enumerate the 100-best networks for real-world datasets with 17 variables [22]. Using heuristic search methods, we can enumerate the $1,000$-best networks for real-world datasets with 23 variables, which is the current state-of-the-art [2]. In this paper, we show how to extend the reach of such systems further, allowing us to enumerate structures for datasets with 29 variables. Each of these advances is quite significant, when we consider how quickly the search space grows, as we increase the number of variables.[1]

More specifically, we propose techniques that can greatly reduce the search space of Bayesian network structures, by safely eliminating regions of the search space that do not contain any of the k-most likely DAGs. By exploiting properties of the popular MDL score for Bayesian networks, we identify an upper bound on the number of parents that a node can have, in any of the k-best structures. Any structure enumeration algorithm that can incorporate such a bound (including all of the aforementioned approaches) can benefit from the techniques that we propose. In fact, our bounds generalize those

☆ This paper is part of the Virtual special issue on the Eighth International Conference on Probabilistic Graphical Models, Edited by Giorgio Corani, Alessandro Antonucci, Cassio De Campos.

* Corresponding author.
E-mail addresses: eyjchen@cs.ucla.edu (E.Y.-J. Chen), darwiche@cs.ucla.edu (A. Darwiche), aychoi@cs.ucla.edu (A. Choi).

[1] For n variables, there are $O(n! \cdot 2^{\binom{n}{2}})$ BN structures. More precisely, for $n = 17, 23$ and 29, there are $6.27 \cdot 10^{52}$, $6.97 \cdot 10^{94}$ and $2.51 \cdot 10^{148}$ structures, respectively; for more on counting Bayesian network structures (and labeled DAGs), see https://oeis.org/A003024.

https://doi.org/10.1016/j.ijar.2017.10.023
0888-613X/

proposed for the problem of learning a single optimal structure [19,21,10]. Such bounds are broadly used in the literature, and the scalability of modern structure learning algorithms depend critically on such bounds.

This paper is organized as follows. In Section 2, we review score-based structure learning and the MDL score. In Section 3, we propose our techniques for pruning the search space for the purposes of enumerating the k-best Bayesian network structures. We evaluate our approach empirically in Section 4, and conclude in Section 5. Proofs are provided in the Appendix.

2. Technical preliminaries and related work

In this section, we first review the score-based structure learning of Bayesian networks, and the problem of enumerating the k-best structures. We then review the MDL score, and prior works that have exploited the MDL score to prune the search space of Bayesian network structures.

We use upper case letters (X) to denote variables and lower case letters (x) to denote their values. Variable sets are denoted by bold-face upper case letters ($\mathbf{X}$) and their instantiations by bold-face lower case letters ($\mathbf{x}$). We use $|X|$ to denote the number of values of a discrete variable X, and $|\mathbf{X}|$ to denote the number of variables in a set $\mathbf{X}$. Generally, we will use X to denote a variable in a Bayesian network and $\mathbf{U}$ to denote its parents. We refer to a variable X and its parents $\mathbf{U}$ as a family, which we denote by $X\mathbf{U}$.

2.1. Score-based structure learning

Score-based approaches for learning the structure of a Bayesian network are based on searching for a DAG that minimizes a given scoring metric, which generally rates the quality of a DAG based (in part) on how well a given structure fits a given dataset $\mathcal{D}$ (which is typically complete). Structure scores often decompose into a sum of local scores, over the families $X\mathbf{U}$ of the DAG:

$$\text{score}(G \mid \mathcal{D}) = \sum_{X\mathbf{U}} \text{score}(X\mathbf{U} \mid \mathcal{D}). \tag{1}$$

For example, MDL and BDeu scores are decomposable (note that we negate such scores as needed to obtain minimization problems). For more on score-based structure learning, see, e.g., [9,14,15].

2.2. Learning the k-best structures

In the problem of enumerating the k-best DAGs, we simply want to find k different DAGs whose scores are the smallest. Enumerating such DAGs can provide a number of insights about a dataset, beyond learning just a single best DAG. First, the single best DAG may be (Markov) equivalent to other equally good DAGs, or otherwise, there may be other DAGs with comparable scores. In either case, we would like to be aware of all such DAGs. Next, if the score that we use corresponds to the probability of a DAG given the data, then enumerating the k-best DAGs gives us a (truncated) view of the posterior over DAGs. If the aggregate probability of the k-best DAGs consumes most of the available probability, then we can further say that the remaining DAGs are unlikely to be relevant. By enumerating the k-best DAGs, we can also look for the structural features that are prominent in the most likely DAGs, as a more tractable approach to Bayesian model averaging; see, e.g., [22,4].

For a concrete example, consider Fig. 1 where we have enumerated all 25 DAGs learned from a dataset $\mathcal{D}$ over 3 variables: B, D and L. The dataset $\mathcal{D}$ was simulated from a known Bayesian network, and hence we know the ground-truth structure: $B \rightarrow D \rightarrow L$. First, each DAG has been ranked by its probability (using its normalized MDL score, which corresponds to the BIC score). DAGs with the same rank and score are grouped together (in this case, each set corresponds to a Markov equivalence class). We note that the top set of two DAGs together have 66.28% probability, although there is a second set of three DAGs which are also relatively likely at 33.60% probability. The remaining 20 DAGs (including the ground truth DAG $B \rightarrow D \rightarrow L$) are relatively unlikely, given the dataset $\mathcal{D}$. Note that our dataset $\mathcal{D}$ is relatively small in this case, with only 100 examples. In Fig. 2, we enumerated all DAGs from a dataset $\mathcal{D}$ over 1,000 examples. Here, the ground-truth DAG is now among the 3 most likely DAGs, which together have 98.01% probability. In Fig. 1, there is a 99.9990% probability that there is an edge connecting variables D and L; in Fig. 2, the probability that there is no such edge is essentially negligible.

One of the first approaches for enumerating the k-best DAGs was based on dynamic programming (DP) [22], and was an extension of a DP-based approach for learning a single optimal DAG [13,17,16]. Another approach encodes the enumeration problem as a series of integer linear programming (ILP) problems [8]. The first ILP problem encodes the problem of finding a single optimal DAG [12,6,8]. This solution is then eliminated by adding a constraint to the ILP problem. The solution of the new ILP problem gives us the second best DAG. We repeat this process until each of the k-best DAGs is enumerated.

The current state-of-the-art for enumerating the k-best DAGs is based on heuristic search methods such as A*, which was previously observed to be orders-of-magnitude more efficient than the above approaches based on DP and ILP [2]. It is based on navigating a seemingly intractable search space over all DAGs. The complexity of this search can be mitigated, however, by exploiting an oracle that can find a single optimal DAG. This search space, called the BN graph, can also be used to learn Bayesian network structures with non-decomposable priors and constraints [2].