Fuzzy Systems and Data Mining IX
A.J. Tallón-Ballesteros and R. Beltrán-Barba (Eds.)

doi:10.3233/FAIA231085

Learning of Bounded Treewidth Bayesian Networks via A-kg

Ronghao Su [a] and Yungang Zhu [a,1]
[a] *College of Computer Science and Technology, Jilin University, Changchun, China*

Abstract. Bounded-treewidth Bayesian networks can reduce overfitting and exact inference complexity. Several known methods learn bounded treewidth Bayesian networks by learning from k-trees. However, they adopt an approximate method instead of an accurate method. This work presents an accurate algorithm called A-kg for learning bounded treewidth Bayesian networks. Our approach consists of two parts. The first part is an accurate algorithm that learns Bayesian networks with high BIC scores, which measures the Bayesian network's quality. In the second part, we adopt the greedy strategy to perform parent set selection efficiently. A-kg achieves better performance compared to some approximate solutions in small domains.

Keywords. Bayesian network, Bounded treewidth, A-kg, BIC

1. Introduction

Bayesian networks are directed graphs widely used to represent the joint probability distribution on multivariate domains and achieve excellent performance in fields such as prediction, inference, diagnosis, decision risk, and reliability analysis.

Learning a Bayesian network refers to inferring its structure from data, a work proven to be an NP-hard problem [1]. Bayesian networks are usually used for inference, such as calculating the posterior probability of some variable given some evidence or finding the mode of the posterior joint distribution. Those inferences are NP-hard to compute even approximately [2]. To make efficient inferences, Bayesian networks need to have small treewidth, assuming exponential time hypothesis (ETH).

The learning methods of Bayesian networks can be divided into accurate and approximate. Yuan found an accurate method to learn Bayesian networks. The method formulated the learning Bayesian network as a shortest path-finding problem and used an A* search algorithm to approach the problem [3]. Di proposed a BN structure learning algorithm based on dynamic programming, which integrated improved MMPC and MWST [4]. In the A* search algorithm, Wang improved the simple heuristic and the static k-cycle conflict heuristic to adapt to ancestral constraints [5]. However, the Bayesian networks learned by these methods are not unbounded treewidth.

For the existing accurate algorithm, in the worst case, the time complexity is the exponential level of the treewidth [6]. So it is necessary to limit the treewidth of Bayesian networks.

[1] Corresponding Author,Yungang Zhu, College of Computer Science and Technology, Jilin University, Changchun ,China; E-mail: zhuyungang@jlu.edu.cn

In recent years, many methods have been proposed to limit the treewidth of Bayesian networks. Ramaswamy combined heuristic BN structure learning algorithms with the recently introduced MaxSAT-powered local improvement method [7]. Ramaswamy also proposed an approach whose key was applying an exact method locally, to improve the score of a heuristically computed BN [8]. Xu proposed a Bayesian network structure learning approach based on full permutation and extensible ordering-based search [9]. Scanagatta proposed k-greedy, k-A*, and k-max algorithms [10]. The researchers used the idea of searching directly for high-quality K-trees and proposed a sampling variable order to learn the optimal DAG. They initialized (k+1) cliques iteratively by adding other variables to the k-clique of the current graph K-tree, establishing the DAG greedily. This method is different from the previous one. And there is no need to learn DAG from a given K-tree. It samples a variable order rather than a tree, resulting in a substantial reduction in search space. There is no need to learn DAG from a given K-tree, and it samples a variable order rather than a tree, resulting in a substantial reduction in search space The difference between k-greedy and k-A* is the way of selecting parent sets. K-greedy selects parent sets with the highest score, while k-A* formulates it as the shortest path-finding problem and solves the problem by the A* algorithm. K-max can learn Bayesian networks from incomplete data sets. All of them can get Bayesian networks with bounded treewidth. However, they are an approximate method that gets lower scores than the accurate method.

This paper presents a new accurate algorithm called A-kg for score-based Bayesian network learning with bounded treewidth. A-kg algorithm consists of two parts: parent identification and structure optimization. In the part of structure optimization, we formulate a learning Bayesian network as a shortest path-finding problem and use an A* search algorithm to approach the problem. This is an accurate method. In the part of parent identification, we draw on the idea of the k-greedy algorithm which selects parent sets by greedy strategy. Here, we can get Bayesian networks with limited treewidth. A-kg is proven to learn a Bayesian network with bounded treewidth and achieve higher scores than traditional algorithms. To test our methods, we compare A-kg and some other algorithms. We compare them with the BIC score. Moreover, we compare them on data sets with different treewidth.

2. Treewidth and K-tree

2.1. Treewidth

The treewidth represents the extent to which a graph resembles a tree. Assuming G (V, E) is an undirected simple graph, the tree decomposition H of graph G is composed of a subset $Yt \subseteq V$ associated with each node of tree T. Tree T and subset $\{Yt: t \in T\}$ should satisfy the following three conditions. The first condition is Eq. (1).

$$\cup \left(Y_t : t \in T\right) = V \tag{1}$$

Eqs. (1) means that the nodes contained in subsets Yt. cover all nodes of graph G, or that each node of graph G belongs to at least one subset Y_t. The second condition is that for each edge $e \in E$ of graph G, there is at least one subset Y_t containing two endpoints of e. The third condition is that if n1, n2, and n3 are the three nodes of tree T,

where n2 is on the path from n1 to n2, and node v of G belongs to Y_{n1} and Y_{n3}, then v will belong to Y_{n2}.

The width of a tree decomposition is equal to max ($|Y_t|$: $t \in T$) -1 where $|Y_t|$ is the number of vertices in $|Y_t|$. The treewidth of H is the minimum width among all possible tree decompositions of G.

2.2. K-tree

The k-tree is the largest graph with a treewidth of k, and every graph with a treewidth ≤ k is a subgraph of some k-trees.

The family of k-trees is defined inductively as follows:(1)A (k+1)-clique is a k-tree. (2) If G=(V, E) is a k-tree and $C \subseteq V$ is a k-clique, then the graph obtained by adding a new vertex v and an edge u−v for each $u \in C$ is a k-tree.

3. A-kg

The main framework of A-kg is the A* search algorithm. By combining the idea of the k-greedy algorithm, A-kg can learn the Bayesian network with bounded treewidth. The Bayesian network obtained by this algorithm is treewidth bounded because each node has a parent set which must be a subset of a k-clique. At the same time, A-kg adopts a greedy strategy to select parent sets. The parent sets are chosen to have the highest score. The following is the definition of A* search and k-greedy.

3.1. A search*

The basic idea of this algorithm is to formulate learning optimal Bayesian networks as a shortest path-finding problem. A * search algorithm starts from an empty set and searches until all nodes are found. The shortest path among all possible paths corresponds to the global optimal Bayesian network. The scoring function is MDL. A-kg uses BIC score. Let U be a node in Figure 1 and V be all nodes, then the heuristic value h(U) is represented by Eqs. (2).

$$h(U)=\sum_{x\in V\setminus U} \text{BestScore}(X, V\setminus\{X\}) \quad =\sum_{x\in V\setminus U} \text{BestMDL}(X, V\setminus\{X\}) \tag{2}$$

The arc from U to $U \cap \{X\}$ in the figure represents the generation of subsequent nodes by adding a new variable {X} to the existing variable set U; The cost of an arc is equal to the cost of selecting a parent set for X from U. The cost is represented by Eqs. (3).

$$\text{BestScore}(X, U)=\min_{P_{A_X}\subseteq U}\text{score}\left(X\middle|P_{A_X}\right) =\min_{P_{A_X}\subseteq U}\text{MDL}\left(X\middle|P_{A_X}\right) \tag{3}$$

3.2. K-greedy

K-greedy first samples a variable order and uses the first （k+1） variable to establish a k-tree. *The accurate* learning algorithm is taken to learn the best DAG on the same k+1 variable. Then, the algorithm iteratively add*s* each remaining variable. The parent set*s* of

this variable are constrained to the k-clique (or subset) in *k*-tree. The selected parents set has the highest score. This results in a new DAG. To update the k-tree, the algorithm links variables to the same k-*clique*. Assuming the sampling variable order is<v1, v2, v3 >, Figure 1 illustrates the process of the algorithm.

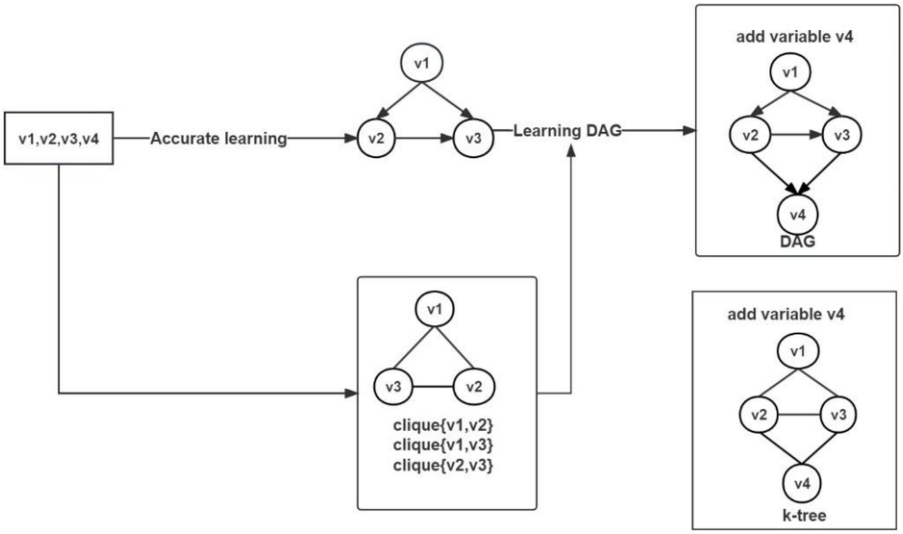

Figure 1. The process of k-greedy.

In the A-kg, we construct a k-clique like the k-greedy algorithm in the process of learning the Bayesian network. The parent set with the highest scoring is chosen for the variable. The parent set of the added node is a subset of k cliques. When connecting this node with the parent set, the maximum clique obtained is also k+1 clique, and the treewidth is still not greater than k. In this way, the learned Bayesian networks are bounded by treewidth. Algorithm 1 shows the process of A-kg.

Algorithm 1 A-kg.

```
1: procedure A-kg
2:     Q1 ← ∅;              // Q1 is open queue
3:     while Q1 is not empty do
4:         S ← H(Q1);   // get the header of Q1
5:         if S contains all variables then
6:             end procedure;
7:         end if
8:         Q2 ← S;       // place S in closed queue(Q2)
9:         for v do       // v is a variable that is not present in S
10:            S1 ← P(v)+S;       // select parent set for v
11:        end for
12:        if Q2 contains S1 & the score of S1 is lower then
13:            discard S1;
14:        else
15:            Q1 ← S1;
16:        end if
17:    end while
18: end procedure
```

In Algorithm 1, the open queue is a priority queue, with BIC as the scoring standard, used to store the states that need to be traversed. The closed queue is a priority queue, with BIC as the scoring standard, used to store the searched status.

4. Experiment

In our experiment, we use an indicator which is the difference between the BIC scores (△BIC) of the DAG [11].

Assuming that there are two graphs G1 and G2, the value of △BIC is the BIC scores of the G1 minus the BIC scores of the G2. A positive △BIC means that G1 is better than G2. Otherwise, it's the opposite. For example, a △BIC that is greater than 2 and less than 6 implies positive evidence in favor of G1. A △BIC that is greater than 10 implies extremely positive evidence in favor of G1. The value of △BIC can be interpreted according to Table 1. We only present the case of positive △BIC. If the value of △BIC is negative, then the evidence is negative.

Table 1. Meaning of different △BIC.

	△BIC<2	**2<△BIC≤6**	**6<△BIC ≤10**	**△BIC >10**
G1vsG2	neutral evidence	positive evidence	strongly positive evidence	extremely positive evidence

4.1. Comparison

We use 4 data sets to compare k-greedy, k-max, k-A* and A-kg. Table 2 shows the details of these data sets. In Table 2,n represents the number of variables and d represents the number of data points.

Table 2. The 4 data sets used in the experiments.

Name	**n**	**d**
Sachs	11	5000
Survey	6	5000
Msnbc	17	58265
Child	20	5000

We start the experiment by calculating the value of △BIC between A-kg and other algorithms. The experiments were run on a computer with 8 cores, a memory limit of 8GB, a time limit of 10 hours, and a maximum number of parents of three**.** The △BIC achieved by different methods is given in Table 3.

Table 3. △BIC between A-kg and other algorithms.

△BIC	**k-greedy**	**k-max**	**k-A***
△BIC >10	0	0	0
6<△BIC ≤10	0	0	0
2<△BIC≤6	2	3	2
0≤△BIC<2	2	1	2
− 6≤△BIC<0	0	0	0
− 10≤△BIC< − 6	0	0	0
△BIC< − 10	0	0	0

The results show that A-kg gets more BIC scores than other algorithms. Take △BIC between A-kg and k-max for example, there are 3 △BICs whose values are greater than

2 and less than 6. According to Table 1, these △BICs mean positive evidence for the Bayesian network learned by A-kg over the Bayesian network learned by k-max. And there is a △BIC between A-kg and k-max whose value is greater than 0 and less than 2. According to Table 1, this △BIC means neutral evidence for the Bayesian network learned by A-kg over the Bayesian network learned by k-max. In the same way, there are 2 △BICs between A-kg and k-greedy which mean positive evidence for the Bayesian network learned by A-kg over the Bayesian network learned by k-greedy, and 2 △BICs between A-kg and k-greedy which mean neutral evidence for the Bayesian network learned by A-kg over the Bayesian network learned by k-greedy. The △BICs between A-kg and k-A* are the same as the △BICs between A-kg and k-greedy.

We further compare A-kg and k-A* under various treewidth. The results are presented in Table 4.

Table 4. Comparison between A-kg and k-max with the treewidth k ∈{3, 5, 7}.

A-kg vs k-max	Treewidth		
	3	***5***	*7*
Extremely positive	0	0	0
Strongly positive evidence	0	0	0
Positive evidence	2	2	2
Neutral evidence	2	2	2
Negative evidence	0	0	0
Strongly negative evidence	0	0	0
Extremely negative evidence	0	0	0

In most cases, there is a piece of positive evidence for the model learned by A-kg over the model learned by k-A*.

5. Conclusions

This paper presents a new algorithm (A-kg) for learning Bayesian networks with bounded treewidth. We compare A-kg and some of the other algorithms. The results show that A-kg gets better scores and finds better structures than its competitors. Furthermore, we compare A-kg and k-max under various treewidth. The results also show that A-kg achieves a better score.

References

[1] Chickering DM. Learning Bayesian Networks is NP-Complete. Learning from Data.1996;112: 121-130.

[2] Roth D. On the hardness of approximate reasoning. Artificial Intelligence.1996;82: 273-302.

[3] Yuan C, Malone B. Learning Optimal Bayesian Networks: A Shortest Path Perspective. Journal of Artificial Intelligence Research. 2013; 48: 23-65.

[4] Di RH, Li Y, Li TP. Dynamic Programming Structure Learning Algorithm of Bayesian Network Integrating MWST and Improved MMPC. Mathematical Problems in Engineering. 2021;2021(53):1-17.

[5] Wang ZD, Gao XG. Learning Bayesian networks using A* search with ancestral constraints. Neurocomputing. 2021; 451: 107- 124.

[6] Kwisthout J, Bodlaender HL, Gaag vr, Linda C. The Necessity of Bounded Treewidth for Efficient Inference in Bayesian Networks. Frontiers in Artificial Intelligence and Applications. 2010; 215: 237-242.

[7] Ramaswamy VP, Szeider S. Learning Fast-Inference Bayesian Networks. In: 35th Conference on Neural Information Processing Systems. 2021. p. 34-46

[8] Ramaswamy VP, Szeider S. Turbocharging Treewidth-Bounded Bayesian Network Structure Learning. In: 35th AAAI Conference on Artificial Intelligence. 2021. p. 3895-3903.

[9] Xu RH, Liu SH. PEWOBS: An efficient Bayesian network learning approach based on permutation and extensible ordering-based search. Future Generation Computer Systems. 2022; 128: 505-520.

[10] Scanagatta M, Corani G, Campos CP, Zaffalon M. Learning Treewidth-Bounded Bayesian networks with thousands of variables. In:30th Annual Conference on Neural Information Processing Systems.2016.p.1470 – 1478.

[11] Lv YL, Miao JZ. BIC-based node order learning for improving Bayesian network structure learning. Frontiers of Computer Science. 2021; 15(6): 2095-2228.