Heuristics for Initializing Order-Based Bayesian Network Structure Learning

Walter Perez Urcia
and
Denis Deratani Mauá

Universidade de São Paulo, Brasil
wperez@ime.usp.br,denis.maua@usp.br

Abstract. A popular and effective method for learning Bayesian network structures is to perform a greedy search on the space of variable orderings followed by an exhaustive search over the restricted space of compatible parent sets. Usually, the greedy search is initialized with a randomly sampled order. In this article we develop heuristics for producing informed initial solutions to order-based search based on the feedback arc set problem.

Categories and Subject Descriptors: I.2.6 [**Artificial Intelligence**]: Learning

Keywords: Bayesian networks, machine learning, local search

1. INTRODUCTION

Bayesian Networks are models that represent efficiently probability distributions between the atributes of a data set. They are defined by two components:

—A directed acyclic graph (DAG), where the nodes are the atributes of the data set and the edges are the independence relationship between the atributes.
—The conditional probability distributions between the atributes and their parents. These are defined by the network's structure.

Formally, a bayesian network is defined by:

$$G = (V, E), where\ V = \{X_1, X_2, \ldots, X_n\}$$

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid Pa_G(X_i))$$

where $Pa_G(X_i)$ is the set of atributes that are parents of X_i.

This definition shows that having less parents for an atribute helps in doing less computations in order to use the network for prediction or inference, but the problem of learning Bayesian networks from data is a NP-hard problem [David M. Chickering 2004]. For this reason, the common approach to solve this problem is to use heuristic search methods, commonly using a scoring function, like the bayesian information criterion (BIC) [Cover and Thomas 1991] or the minimum description length (MDL) [Wai Lam 1994].

In this article we focus on the Greedy Search method that gets a random initial solution and do a local search between the space of orderings of the atributes. We also propose a new method for generating an informed initial solution which is based on the Feedback Arc Set Problem (FASP).

The article is structured as follows: We begin in Section 2 explaining the Greedy Search algorithm. In Section 3, we describe the new algorithm for generating initial solutions. Section 4 shows the experiments using both approaches and comparing them (in time and scoring) with multiple data sets. Finally, in Section 5 we give some conclusions about the two approaches.

2. LEARNING BAYESIAN NETWORKS

2.1 Definition of the problem

The problem of learning the structure of a Bayesian network is stated as: Given a training data set D, select the network (DAG) G that maximizes the scoring function $sc(G, D)$.
In this article, we will use the bayesian information criterion (BIC) as scoring function that is defined as follows:

$$sc(G, D) = BIC(G, D) = DLL(G) + \frac{\log N}{2} size(G)$$

where

$$DLL(G) = \sum_{i=1}^{n} \sum_{k} \sum_{j} N_{ijk} \log \frac{N_{ijk}}{N_{ij}}$$

$$size(G) = \sum_{i=1}^{n} (|\Omega_i| - 1) \prod_{X_j \in Pa(X_i)} |\Omega_j|$$

Also, N is the number of instances in the data set, n the number of atributes in the data set,N_{ijk} the number of instances that has the values i, j and k (the same way for N_{ij}). Finally, Ω_i is the size of the set of possible values for the attribute i.

Having in consideration all these formulas, we can decompose the score function in order to calculate for one attribute.

$$BIC(X_i, Pa(X_i)) = DLL(X_i, Pa(X_i)) + \frac{\log N}{2} size(X_i, Pa(X_i))$$

Using all the formulas, we can reduce the problem to the following formula:

$$G = \max_{G} \sum_{i=1}^{n} BIC(X_i, Pa(X_i)) = \sum_{i=1}^{n} \max_{P} BIC(X_i, P)$$

In other words, to get a better network, we need to pick the best set of parents P for each attribute. But the main problem with searching P is that we have 2^{n-1} possible set of parents for each attribute, it means, an exponential number of possible sets based on the number of attributes in the data set.

In order to avoid this problem is added a new constraint d, where d is the maximum number of parents for each attribute, so we have only 2^d possible sets.

$$G = \max_{G} \sum_{i=1}^{n} BIC(X_i, Pa(X_i)) = \sum_{i=1}^{n} \max_{|P| \leq d} BIC(X_i, P)$$

2.2 Greedy Search Algorithm

Greedy Search algorithm is a popular and effective solution to the problem of learning the structure of a Bayesian network. Algorithm 1 shows its pseudocode.

Algorithm 1: Greedy Search

```
1   L = Permutation( X_1 , ... , X_n )
2   For a number of iterations K do
3           current_sol = find_order( L )
4           if score( current_sol ) > score( L ) :
5                   L = current_sol
6   Return L
```

A pseudocode for *find_order* is showed below.

Algorithm 2: Find_Order

```
1   For each i do
2           L_i = [ L[ 1 ] , ... , L[ i + 1 ] , L[ i ] , ... , L[ n ] ]
3           if score( L_i ) > score( L )
4                   best = L_i
5   Return best
```

The main idea of the algorithm is to generate an initial solution using a permutation of the attributes. After that, for a number of iterations K, perform swaps between consecutive attributes and calculate their score in order to find a better solution. Finally, return the best solution after all the iterations. In section 4, this algorithm will be used for learning structure of Bayesian networks using multiple data sets.

3. GENERATING INFORMED INITIAL SOLUTIONS

An important step in algorithm 1 is in the first line, where is created a initial solution for the data set that is based on a permutation of the attributes.In this section, we propose another way to generate an initial solution. Basically, the first line will be changed to another method as is showed in Algorithm 3.

Algorithm 3: Modification of Local Search

```
1   L = Informed_solution( X_1 , ... , X_n )
2   For a number of iterations K do
3           current_sol = find_order( L )
4           if score( current_sol ) > score( L ) :
5                   L = current_sol
6   Return L
```

The method for generating an informed solution is as follows:

(1) Find the set of best parents (using the constraint d in previous section) for each attribute
(2) Build a graph using the relations between attributes and their best parents as edges
(3) For each node X in the graph do
   (a) Do a depth-first search (DFS) on the graph using X as root
   (b) Change the edge's directions that made cycles to obtain a DAG
   (c) Calculate the network's score
(4) Choose the best network and return it

In step 1, we find the best parents for each attribute X_i performing a greedy search like follows:

(1) Start with an empty set P
(2) If $|P| = d$, return P

(3) Select $X_k \notin P$ that maximizes $BIC(X_i, P \cup X_k)$ and $i \neq k$

(4) If exists X_k, add X_k to P and go to step 2

With all the sets calculated, we use that relations as edges to build a graph. Notice that this graph can have cycles and a Bayesian network does not have cycles.
The problem to transform an unweighted directed graph with cycles to a DAG is called Feedback Arc Set Problem (or FASP). This problem is NP-Hard, but exists approximations for solving it. A popular one is similar to step 3, but it only choose a random node and do not compare all possible ways. As we need to obtain the best possible initial solution, a loop over the attributes is done and step 3.c is added for comparison between them. Finally, we return the best network.

This modification of Local Search will be used in next section to learn Bayesian networks with multiple data sets.

4. EXPERIMENTS AND RESULTS

4.1 Configuration

Table I shows the characteristics of each data set used in the experiments.

Name	n (#attributes)	N (#instances)
Census	13	45000
Dataset 2	n_2	N_2
Dataset 3	n_3	N_3
Dataset 4	n_4	N_4

Table I: Data sets

Also, the values for each general parameter are given below.

—Training/test percentage instances: 65% / 35%

—Maximum number of parents (d): 4

—Number of iterations in Local Search (K): 100

4.2 Results

After running both algorithms for each data sets, we compare the cost (Data Log-Likelihood) and the CPU Time. Tables II and III show them respectively.

Name	Random Sol.	Informed Sol.
Census	-165350	-14683
Dataset 2	n_2	N_2
Dataset 3	n_3	N_3
Dataset 4	n_4	N_4

Table II: Data Log-Likelihood using each approach

Name	Random Sol.	Informed Sol.
Census	10.2 / 72.14	93.51 / 48.89
Dataset 2	n_2	N_2
Dataset 3	n_3	N_3
Dataset 4	n_4	N_4

Table III: CPU Time (in seconds) for initializing/searching using each approach

5. CONCLUSIONS

We conclude that:

—Generating an informed initial solution gives best networks that taking a random one
—To generate the informed initial solution takes more CPU time that the one in the first approach
—A higher number of attributes implies more CPU time for the second approach with a significant difference compared to the first approach
—With an informed solution, Local Search algorithm needs less CPU time for finding a better network

REFERENCES

Cover, T. M. and Thomas, J. A. *Elements of Information Theory*. Wiley-Interscience, 1991.

David M. Chickering, David Heckerman, C. M. Large-Sample Learning of Bayesian Networks is NP-Hard. 5 (1): 1287–1330, 2004.

Wai Lam, F. B. Learning Bayesian Belief Networks. An approach based on the MDL principle. 10 (4): 31, 1994.