Structure Learning of Bayesian Networks using Genetic Algorithms

Mrinmoy Sarkar, Biniam T. Gebru**, and Abdollah Homaifar****

Abstract— **In this project we investigate the problem of the search for the best Bayesian network structure, given a database of cases, using the genetic algorithm philosophy for searching among alternative structures. We start by assuming an ordering between the nodes of the network structures. This assumption is necessary to guarantee that the networks that are created by the genetic algorithms are legal Bayesian network structures. Next, we release the ordering assumption by using a "repair operator" which converts illegal structures into legal ones. We present empirical results and analyze them statistically. The best results are obtained with an elitist genetic algorithm that contains a local optimizer.**

Index Terms—**Genetic algorithm, structure learning, Bayesian networks**

I. Introduction

BAYESIAN networks (BNs) have become popular with in the AI community as a method of reasoning under uncertainty and quest for causal reasoning. From an informal perspective, BNs are directed acyclic graphs (DAGs), where the nodes are random variables and where the arcs specify the independence assumptions between these variables. After construction, a BN constitutes an efficient device for performing probabilistic inference.

The problem of searching the BN that best reflects the dependence relations in a database of cases is a difficult one because of the large number of possible DAG structures, given even a small number of nodes to connect. In this paper, we present a method for solving this problem of the structure learning of BNs from a database of cases based on genetic algorithms.

The structure of the paper is as follows. In the next section, we introduce the Bayesian networks and we describe the problem of the search of such a network from a database of cases. A brief introduction on genetic algorithms is given in Section 3. In Section 4, we show how genetic algorithms can be used for tackling the problem of the structure learning of BNs. We describe two different approaches, namely with and without assuming an ordering between the nodes of the network. In the latter approach, the offspring constructed by the genetic algorithm are not necessarily BN structures, they may have to be repaired. In both approaches we use a Bayesian approach to measure the fitness of the structures. Empirical results obtained with simulations of the ASIA and ALARM networks are pre-

** Mrinmoy Sarkar is with the NC A&T SU, ACIT & TECHLAV Center (corresponding author: msarkar@aggies.ncat.edu).*

*** Biniam T. Gebru is with the NC A&T SU, ACIT & TECHLAV Center (btgebru@aagies.ncat.edu)*

**** Abdollah Homaifar is with the NC A&T SU, Duke Energy Eminent Professor and Director of ACIT & TECHLAV Center (homaifar@ncat.edu)*

II. Bayesian Networks

A. Overview

A Bayes network is a structure that can be represented as a directed acyclic graph, and the data it contains can be seen from the following two points of view: it allows a compact and modular representation of the joint distribution using the chain rule for Bayes network. Additionally, it describes the conditional independence assumptions between vertices to be observed.

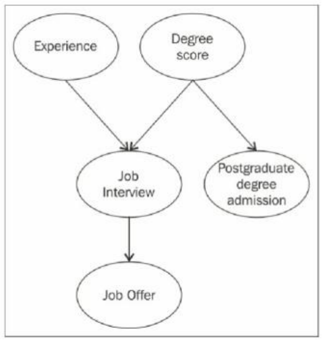


Figure 1: A typical BN

A BN is described by a chain of conditional inter-dependencies:

- General Representation
  $P(E, D, I, J, A) = P(E)P(D|E)P(I|E,D)P(J|E,D,I)P(A|E,D,I,J)$
- Bayesian relation
  $P(E, D, I, J, A) = P(E)P(D)P(I|E,D)P(J|I)P(A|D)$

B. *Reasoning Patterns*

Reasoning in Bayesian network means how the value of probability of one random variable changes with respect to another random variable which value has been observed already in the network. There are three types of reasoning in a Bayesian network: *Causal reasoning, evidential reasoning and Inter-causal reasoning*

Causal Reasoning

Causal reasoning means we observe the prior probability of an event unconditioned by any evidence. We then introduce observations of one of the parent variables. Consistent with our logical reasoning, we note that if one of the parents (equivalent to causes) of an event is observed, then we have stronger beliefs about the child random variable.

Evidential Reasoning

Evidential reasoning is when we observe the value of a child variable, and we wish to reason about how it strengthens our beliefs about its parents.

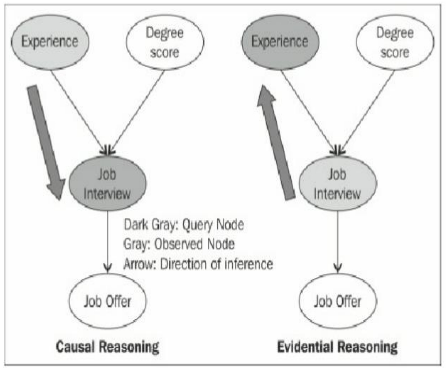


Figure 2: Causal & Evidential Reasoning Methods

Inter-causal Reasoning

Inter-causal reasoning, as the name suggests, is a type of reasoning where multiple causes of a single effect interact.

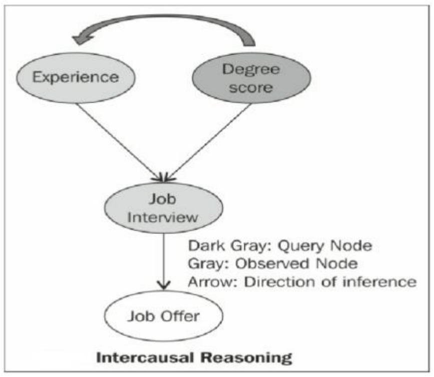


Figure 3: Inter-causal reasoning method

C. *D-Separation & Active Trail*

D-separation means how information flows in a Bayesian network and how random variables are separated from one another in the context of information flow.

No variables observed	Y has been observed
$X \leftarrow Y \leftarrow Z$ ☺	$X \leftarrow Y \leftarrow$ ☹
$X \rightarrow Y \rightarrow Z$ ☺	$X \rightarrow Y \rightarrow Z$ ☹
$X \leftarrow Y \rightarrow Z$ ☺	$X \leftarrow Y \rightarrow Z$ ☹
$X \rightarrow Y \leftarrow Z$ ☹	$X \rightarrow Y \leftarrow Z$ ☺

Table 1: D-separation and information flow

A trail is active if it contains no V-structures in the event that no evidence is observed. In case multiple trails exist between two variables, they are conditionally independent if none of the trails are active. In the previous table the smiley trails are active trail.
In the left figure active trail is shown when no random variable is observed and in the right figure active trail is shown when Ji is observed.

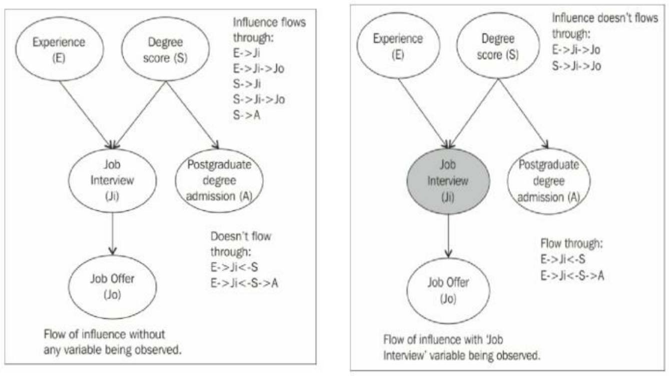


Figure 4 V-structures

III. Structure Learning

The number of possible network structure for n node of a Bayesian network can be found by using the following formula:

$$f(n) = \sum_{i=1}^{n} (-1)^{i+1} \frac{n!}{(n-i)!n!} 2^{i(n-i)} f(n-1)$$

Nodes	Number of DAGs	Nodes	Number of DAGs
1	1	13	$1.9 \cdot 10^{31}$
2	3	14	$1.4 \cdot 10^{36}$
3	25	15	$2.4 \cdot 10^{41}$
4	543	16	$8.4 \cdot 10^{46}$
5	29281	17	$6.3 \cdot 10^{52}$
6	$3.8 \cdot 10^{6}$	18	$9.9 \cdot 10^{58}$
7	$1.1 \cdot 10^{9}$	19	$3.3 \cdot 10^{65}$
8	$7.8 \cdot 10^{11}$	20	$2.35 \cdot 10^{72}$
9	$1.2 \cdot 10^{15}$	21	$3.5 \cdot 10^{79}$
10	$4.2 \cdot 10^{18}$	22	$1.1 \cdot 10^{87}$
11	$3.2 \cdot 10^{22}$	23	$7.0 \cdot 10^{94}$
12	$5.2 \cdot 10^{26}$	24	$9.4 \cdot 10^{102}$

Table 2: The no. of different DAGs that can be generated for a given no. of nodes

General Procedure in Structure Learning

There are two types of structure learning for BN DAGs: namely, constraint based learning and score based learning. In this paper we are going to employ score-based learning and the subsequent discussion will be on the latter.
There are two component of score based learning: a score function and a search procedure.

Score function:

$$\operatorname{dist}\left(P_D(\mathcal{U}), P_S(\mathcal{U} \mid \hat{\boldsymbol{\theta}})\right) = \sum_{\mathbf{x} \in \operatorname{sp}(\mathcal{U})} \left(P_D(\mathbf{x}) - P_S(\mathbf{x} \mid \hat{\boldsymbol{\theta}}_S)\right)^2 .$$

Since a complete network structure can encode any probability distribution, we know that in order to minimize the distance above, we can simply select any complete network structure.
To avoid this problem we could augment the score with a term penalizing model complexity.

$$\operatorname{score}\left(P_D(\mathcal{U}), P_S(\mathcal{U} \mid \hat{\boldsymbol{\theta}})\right) = \operatorname{dist}\left(P_D(\mathcal{U}), P_S(\mathcal{U} \mid \hat{\boldsymbol{\theta}})\right) + c \cdot \operatorname{size}(S)$$

where c is a (user specified) constant used to control the trade-off between model accuracy and model complexity

Properties of a score function:

1. It should balance the accuracy of a structure with the complexity of the structure.
2. It should be computationally tractable to evaluate.

Accordingly, the Baysian Information Criterion (BIC) is defined as a score function:

$$\operatorname{BIC}(S \mid \mathcal{D}) = \log_2 P(\mathcal{D} \mid \hat{\theta}_S, S) - \frac{\operatorname{size}(S)}{2} \log_2(N)$$

If we furthermore assume that the cases are independent given the model, then

$$\operatorname{BIC}(S \mid \mathcal{D}) = \sum_{i=1}^{N} \log_2 P(\mathbf{d}_i \mid \hat{\theta}_S, S) - \frac{\operatorname{size}(S)}{2} \log_2(N).$$

The calculation of the BIC score can be reduced to a counting problem:

$$\operatorname{BIC}(S \mid \mathcal{D}) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log_2\left(\frac{N_{ijk}}{N_{ij}}\right) - \frac{\log_2 N}{2} \sum_{i=1}^{n} q_i(r_i - 1)$$

Where, N_{ijk} denotes the number of cases in the database with X_i in its kth configuration and pa(X_i) in the jth configuration, r_i denote the number of states for variable X_i, q_i denote the number of configurations over the parents for X_i in S (if X_i does not have any parents then $q_i = 1$).

IV. Genetic Algorithms

Recently five approaches of heuristic search have emerged for solutions to combinatorial complex problems: evolutionary algorithms, neural networks, simulated annealing, tabu search, and target analysis. The first two-evolutionary algorithms and neural networks-are inspired by principles derived from biological sciences; and simulated annealing derives from physical science, notably the second law of thermodynamics. Tabu search and target analysis stem from the general tenets of intelligent problem-solving.

Roughly, a GA works as follows: First, the initial population is chosen, and the quality of each of its individuals is determined. Next, in every iteration parents are selected from the population. These parents produce children, which are added to the population. For all newly created individuals of the resulting population a probability near zero exits that they "mutate," i.e., they change their hereditary distinctions. After that, some individuals are removed from the population according to a selection criterion in order to reduce the population to its initial size. One iteration of the algorithm is referred to as a generation. The operators which define the child production process and the mutation process are called the crossover operator and the mutation operator respectively. Both operators are applied with different probabilities named the crossover probability and the mutation probability. Mutation and crossover play different roles in the GA. Mutation is needed to explore new states and helps the algorithm to avoid local optima. Crossover should increase the average quality of the population. By choosing adequate crossover and mutation operators as well as an appropriate reduction mechanism, the probability that the GA results in a near-optimal solution in a reasonable number of iterations increases.

A. Representation of Candidate Solution

Suppose we have three variable X1, X2 and X3 The network shown in the figure can be represented as binary matrix as below:

$$\begin{matrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ \downarrow & \downarrow & \downarrow \end{matrix}$$

The meaning of the representation is X1 has no parent but X2 and X3 both has X1 as their parent. If we expand the matrix in column wise then we get our candidate solution for the given structure. e.g.: 011000000

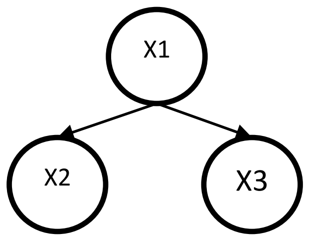


Figure 5: BN

B. GA Operations

Crossover

Suppose we have two candidate solutions given below:
P1 = 000000110
P2 = 001001000
If we choose crossover point at k=6 then,
C1 = 001001110
C2 = 000000000
But C1 is not valid structure
Remark: So, crossover is not a closed operator for this problem.

Mutation

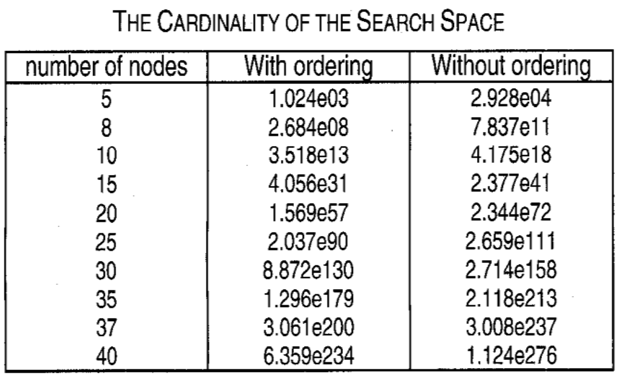
Suppose we have one candidate solution as given below:
P1 = 010001000
Now if we flip the red bit then the resultant child is:
C1 = 01001100
Which is not a valid structure.
Remark: So, mutation is not a closed operator too for this problem.

C. Solution Space for GA Operators

If we consider ordering between nodes the problem we faced before can be solved. Ordering means node Xi can only have node Xj as a parent node if in the ordering node Xj comes before Xi. For instance, suppose we have a order for three node as X2,X1,X3 then X2 will never have any parent, X1 can have X2 as parent and X3 can have X1 and X2 as its parent. For this solution the no. of bits for representing a candidate solution reduces to $\binom{n}{2}$ which is n^2 for general case for n nodes.

Remark: Now, both crossover and mutation are closed operator for this problem.

Solution Space after Ordering



THE CARDINALITY OF THE SEARCH SPACE

number of nodes	With ordering	Without ordering
5	1.024e03	2.928e04
8	2.684e08	7.837e11
10	3.518e13	4.175e18
15	4.056e31	2.377e41
20	1.569e57	2.344e72
25	2.037e90	2.659e111
30	8.872e130	2.714e158
35	1.296e179	2.118e213
37	3.061e200	3.008e237
40	6.359e234	1.124e276

Table 3. Solution Space after ordering operation.
Note that the problem is still NP hard.

V. Implementation/Simulation details

A. Approach

1. Determinate a BN (Structure + conditional probabilities) and simulate it, obtain a database of cases D, which must reflect the conditional independence relations between the variables.
2. Using GA find BN structure B^* , which maximizes the probability P(D|B)
3. Evaluate the fitness of the solutions found

B. Network Under Consideration

The BN used in the simulation sis the ASIA network shown below:

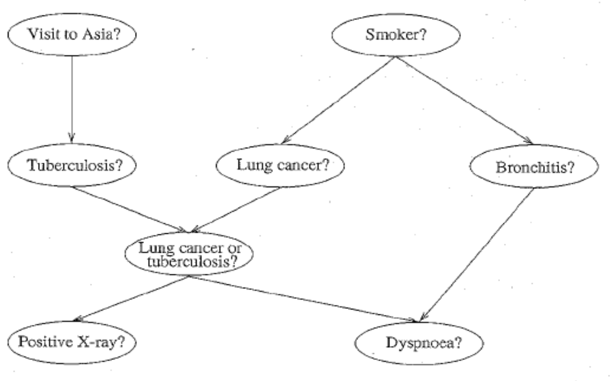


Figure 6: The structure of the ASIA network

C. Simulatin Model

The following simulation model for comparison of initial and induced structures is employed

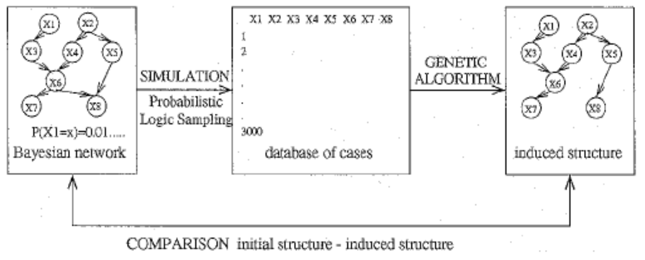


Figure 7: Simulation Model

VI. Results

The following results have been obtained for cancer network with five nodes:

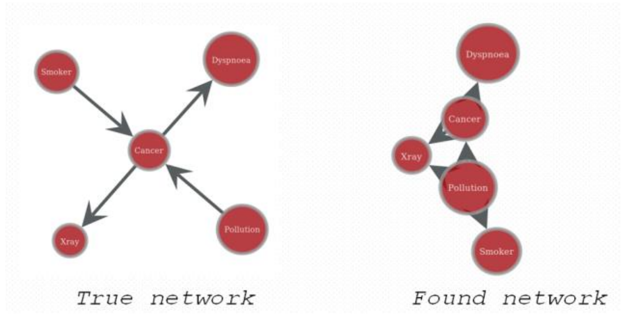


Figure 8: A 5 node cancer network: actual and simulated

Population Size	Data Sample Size	$-logP(D\|B*)$	$-logP(D\|Bo)$	K2 score(B*)	K2 score(B°)
10	500	316.53	499.69	-743.46	-1177.92
10	3000	2074.14	2880.55	-4813.96	-6669.78
50	500	103.97	474.54	-249.20	-1119.81
50	3000	1050.17	2921.04	-2430.56	-6762.44

Table 4: *No. of generation = 100, pc = 0.5, pm=0.01*

Population Size	Data Sample Size	$-logP(D\|B*)$	$-logP(D\|Bo)$	K2 score(B*)	K2 score(Bo)
10	500	235.27	462.42	-554.82	-1091.81
10	3000	2743.67	2901.87	-6339.27	-6719.01
50	500	367.89	490.68	-868.31	-1156.62
50	3000	1201.55	2859.99	-2777.47	-6622.12

Table 5: *No. of generation = 100, pc = 0.9, pm=0.01,*

The following results have been obtained for ASIA network for eight nodes

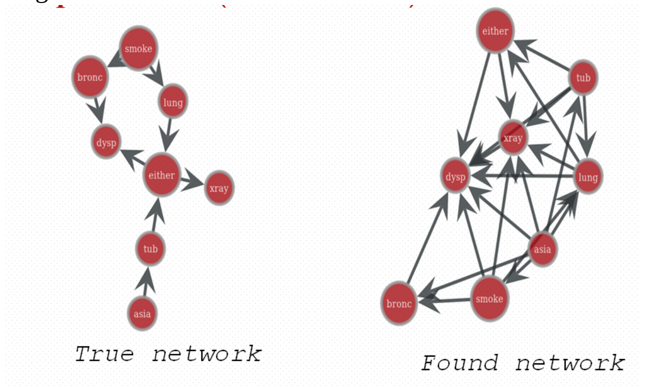


Figure 9: An 8 node cancer network: actual and simulated

Population Size	Data Sample Size	$-logP(D\|B*)$	$-logP(D\|Bo)$	K2 score(B*)	K2 score(B°)
10	500	592.26	513.28	-1426.52	-1230.89
10	3000	2960.26	2964.81	-6922.40	-6894.1545
50	500	398.29	469.03	-963.92	-1128.87
50	3000	2201.76	2983.63	-5161.37	-6937.31

Table 6: *No. of generation = 100, pc = 0.5, pm=0.01*

Population Size	Data Sample Size	$-logP(D\|B*)$	$-logP(D\|Bo)$	K2 score(B*)	K2 score(B°)
10	500	528.94	490.06	-1271.78	-1177.06
10	3000	3074.21	2952.07	-7203.02	-6864.96
50	500	339.72	492.11	-813.26	-1182.66
50	3000	2454.09	3020.86	-5716.59	-7022.89

Table 7: *No. of generation = 100, pc = 0.9, pm=0.01*

VII. Conclusions

Based on the simulation results the following conclisuions are darwn:

- Increasing the data sample size gives better result
- Increasing the no of population doesn't result in significantly improved change in performance (mainly depends on how fit the samples are to the found structure) but the computation becomes slower.
- Increasing the crossover probability gives better result

VIII. References

[1] Pedro Larranaga, et al, *Structure Learning of Bayesian Networks by Genetic Algorithms: A performance analysis of Control Parameters*, IEEE Transactions of pattern Analysis and Machine Intelligence, Vol. 18, No. 9, Sept, 1996

[2] Ben-Gal I., *Bayesian Networks*, Encyclopedia of Statistics in Quality and Reliability, Wiley & Sons, 2007

[3] https://www.bayesserver.com/docs/introduction/bayesian-networks Retrieved on 18th March, 2018