2012 International Conference on Information and Knowledge Management (ICIKM 2012)
IPCSIT vol.45 (2012) © (2012) IACSIT Press, Singapore

Structure Learning of Bayesian Networks Using Heuristic Methods

Alireza Sadeghi Hesar [1 +], Hamid Tabatabaee [2], and Mehrdad Jalali [3]

[1] Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran
[2] Department of Computer Engineering, Ghoochan Branch, Islamic Azad University, Ghoochan, Iran
[3] Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran

Abstract. In this paper, we introduce bayesian artificial networks as a causal modeling tool And analyse bayesian learning algorithms. Two important methods of learning bayesian are parameter learning and structure learning. Because of its impact on inference and forecasting results, Learning algorithm selection process in bayesian network is very important. As a first step, key learning algorithms, like Naïve Bayes Classifier, Hill Climbing, K2, Greedy Thick Thinning are implemented and Are compared based on accuracy and structured network time. Finally, the best of learning algorithm will be proposed. We work with a database of observations (monthly rainfall) measured for the years 1985-2010 in a network of 22 stations in the (Rzavi, Shomali And Jonoubi) Khorasan provinces and with the corresponding gridded atmospheric patterns generated by a numerical circulation model.

Keywords: Bayesian Network, Learning Algorithms, Structure Learning, Parametr Learning, Meteorological Databases

1. Introduction

A Bayesian network or BN is a model that reflects the states of real world and describes how those states are related together by probabilities. Using this framework, the inherent structure of different processes can be interpreted. Although Bayesian Networks provide a means for inference but finding the structure of the networks remains an NP-hard problem. The reason for this is that there is an enormously large number of ways in which the network nodes can be linked to each other. In order to mitigate this problem, a number of algorithms have been proposed. Like, the Naive Bayes Classifier, K2, Local K2, Greedy Thick Thinning or GTT algorithms and etc. The main purpose of this paper to determine the algorithm which produces the Bayesian network with the highest predictive accuracy, and is constructed in the least amount of time. The first part of the article provides a brief introduction of bayesian networks. Then, experimental data sets and methods of normalization is reviewed. In the next section, the main bayesian learning algorithms is introduced and inference methods of them is descripted. In the last section, Results related to the implementations and comparison of algorithms is expressed [1][2].

2. Bayesian Networks And Available Data

A bayesian network is a directed acyclic graph or DAG with a conditional probability distribution for each node. The DAG structure of such networks contains nodes representing domain variables, and arcs between nodes representing probabilistic dependencies. On constructing Bayesian networks from databases, we use nodes to represent database attributes. Because the absence of an edge between two nodes implies conditional independence, the probability distribution of a node can be determined by considering the distributions of its parents. In this way, the joint probability distribution for the entire network can be

+ Corresponding author. Tel.: +985116614278;
E-mail address: alireza.sadeghi89@yahoo.com.

specified. This relationship can be captured mathematically using the chain rule in Equation 1.

$$P_r\,(y_1, y_2, \dots, y_n) = \Pi_{(i=1\ to\ n)}\, P\,(y_i/\pi_i) \qquad (1)$$

In general terms, this equation states that the joint probability distribution for node x is equal to the product of the probability of each component X_i of X given the parents of X_i. Figure 1 shows a simple bayesian network for a diagnosis problem when the all variables are discrete. In this work we consider the Khorasan provinces (North Khorasan, South Khorasan and Razavi Khorasan) in Iran as the geographical area of interest, and use monthly data (rainfall) from a 22 weather stations network (see Figuer. 2) provided by the Iran weather service. The data covers the period from 1985 to 2010 and is representative of the local climatology and synoptic of this area. The variables are considered continuous for some applications, but are also quantized for other applications. Rainfall is quantized into four different states, 0=dry, 1=light rain, 2=moderate rain, 4=heavy rain. According to the thresholds 0, 2, 10 and 20 mm/day, respectively (for instance, the event “heavy rain” corresponds to rainfall > 20 mm) [1].

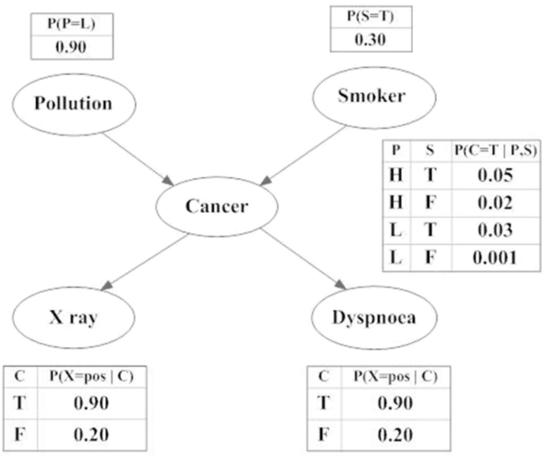


Fig. 1: Simple Example, diagnosis using BN

Fig. 2: Indicating 22 Stations In The Khorasan Provinces

3. Bayesian Structure Learning Algorithms

Bayesian network structure learning algorithms can be grouped in two categories including constraint based algorithms and score based algorithms. Constraint based algorithms learn the network structure by analyzing the probabilistic relations entailed by the Markov property of Bayesian networks with conditional independence tests and then constructing a graph which satises the corresponding d-separation statements. The resulting models are often interpreted as causal models even when learned from observational data. Constraint-based algorithms are all based on the inductive causation (IC) algorithm, which provides a theoretical framework for learning the structure causal models. But Score based algorithms assign a score to each candidate Bayesian network and try to maximize it with some heuristic search algorithm. Greedy search algorithms are a common choice, but almost any kind of search procedure can be used. Score based algorithms on the other hand are simply applications of various general purpose heuristic search algorithms, such as hill climbing, tabu search, simulated annealing and various genetic algorithms. The score function is usually score equivalent, so that networks that de_ne the same probability distribution are assigned the same score [3].

4. Constraint Based Learning

4.1. Naïve Bayes Classifier

The constraint based learning algorithms attempt to isolate the dependency between variables, while not considering the effect of other variables on their relationship. An example of the constraint based learning is the Naive Bayes algorithm. A Naïve Bayes Classifier is a casual modeling or CM with only one parent which forms the classifier node (see Figure 3). All the other nodes within the network form its children. This unique structure is based upon the principle that all the child nodes are independent of each other, given that we can identify the state of the parent, or classifier node. This is not always a wise option as the network structure may not accurately represent the real word domain, and therefore it will be unable to fully capture the variable relationships. However, there are advantages to using this structure such as a shorter construction time and most notably a shorter inference time [2].

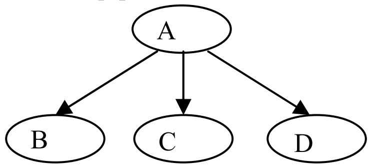


Fig. 3: A Naive Bayes Network with node A As The Classifier

5. Score Based Learning

5.1. K2 Algorithm

The K2 algorithm (see Figuer 4) is greedy algorithm, which obtains the best structure through a iterative process among all possible arangement. In order to compensate for this, the algorithm has to iterate through many structures to ensure that the best scoring one is found. To score the network, the K2 algorithm requires that the order of the variables be known before scoring can start. This constraint prevents cycles from being introduced into the DAG. The scoring function of K2 algorithm for i nodes is in equation 2. The final score of network will obtains by multiplying the individual score of nodes.

$$f(\mathrm{i}, \pi_{\mathrm{i}}) = \prod_{j=1}^{|\Phi_i|} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} \alpha_{ijk}! \tag{2}$$

i is the current node, r_i is the number of states, π_i is the parent of, $|\varphi_i|$ is the number of values within the CPT of, α_{ijk} is The number of cases in the dataset in which has its k_{th} value and have their j_{th} value in CPT and N_{ij} is sum of α_{ijk} for each state of i.

```
K2 Algorithm
1. procedure K2;
2. {Input: A set of n nodes, an ordering on the nodes,
an upper bound u on the
3. number of parents a node may have,
and a database D containing m cases.}
4. {Output: For each node, a printout of the parents of the node.}
5. for i:= 1 to n do
6. πi := φ;
7. Pold := f(i, πi);
8. OKToProceed := true;
9. While OKToProceed and | πi | < u do
10. let z be the node in Pred(xi) - πi that maximizes f(i, πi ∩{z});
11. Pnew:= f(i, πi ∩{z});
12. if Pnew > Pold then
13. Pold := Pnew;
14. πi := πi ∩ {z};
15. else OKToProceed := false;
16. end {while};
17. write(“Node: “, xi, “ Parent of xi: “,πi);
18. end {for};
19. end k2;
```

Fig. 4: The pseudo code for the K2 algorithm

5.2. Hill Climbing And Iterated Hill Climbing

The idea of a hill climbing search algorithm (see Figuer.5) is to generate a model in a step by step fashion by making the maximum possible improvement in an objective quality function at each step. Initialize with a network structure, possibly random, evaluate the change in the score for all arc changes on this network and choose the one that has the maximum change. Continue this process until no more arc changes increase the score. This algorithm generally sticks into local maxima. Various other optimization techniques, such as iterated hill-climbing try to overcome this problem. Iterated hill climbing apply local search until local maximum (see Figuer. 6). Randomly perturb the structure and repeat the process for some manageable number of iterations [4]. The construction time results in Figure 7 show that for every algorithm there is an increase in the construction time when additional neighbours are included. Figure 8 shows that the hill climbing and iterated hill climbing algorithms produce networks which have the highest acurracy when using 4 parent [4].

```
Hill Climbing Algorithm
1. Procedure HillClimbing ;
2. Generate a solution ( s' );
3. Best = S';
4. Loop
   5. S = Best;
   6. S' = Neighbors (S);
   7. Best = SelectBest (S');
8. Until stop criterion satisfied
9. End
```

Fig. 5: The Pseudo Code For The Hill Climbing

```
Iterated Hill Climbing Algorithm
1. Procedure HillClimbing ;
2. Generate a solution ( s' );
3. Best = S';
4. Loop
   5. S = Best;
   6. S' = Neighbors (S);
   7. Best = SelectBest (S');
 8. IF there is no changes in Best solution THEN
 9. Jump to new state in state space
10. Until stop criterion satisfied
 11.End
```

Fig. 6: The pseudo code for the Iterative Hill Climbing

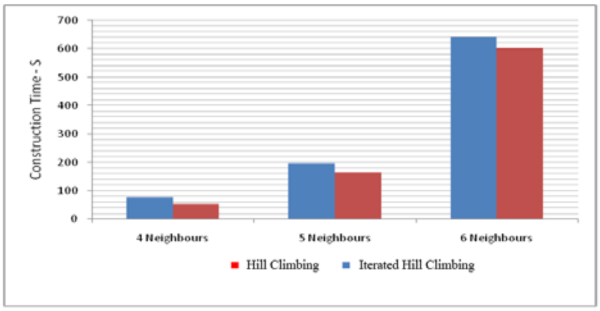


Fig. 7: Construction Time

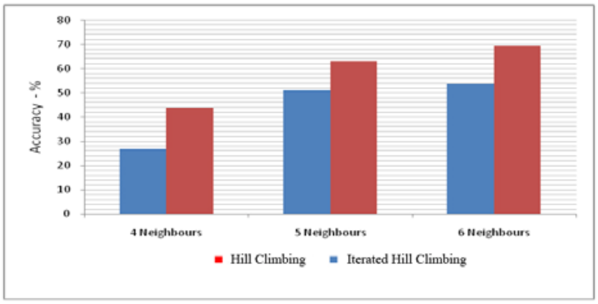


Fig. 8: Acurracy Prediction

5.3. Greedy Tick Thinning

The Greedy Tick Thinning algorithm starts with an empty gtaph and repeatedly adds the arc that maximally increases the bayesian metric until no arc addition will result in an increase. Then it repeatedly removes arcs until no arc deletion will result in an increase in the bayesian metric. Thus, the model produced from this algorithm is the model for which the data is most likely to be observed.The bayesian metric is computed using equation 3, The bayesian metric for GTT is detailed as follows:

$$P\left(D\middle|BN\right)=\Pi_{i=1}^{n}\ \Pi_{j=1}^{\varphi}\frac{\Gamma\left(N_{ij}\right)}{\Gamma\left(N_{ij}+N_{ij}\right)}\ \Pi_{k=1}^{n}\frac{\Gamma\left(N_{ijk}+N_{ijk}\right)}{\Gamma\left(N_{ijk}\right)} \tag{3}$$

n is the number of variables, r_i is the number of states of variable I, N_{ijk} is the number of instance where variable i takes on states k when its parent is in states j, N_{ij} = sum of N_{ijk} N_{ijk} is the Dirichlet exponent of , the probability that variable i is in state k given the parents of i are in state j.

6. Exprimental Results

In our experiments, we used meteorological data bases for testing. Our data set is the monthly rainfall data for 15 years that contains 3960 records. 22 Weather stations in area of study were considered as variables. Network arcs represents a relationship between two stations .This means that the network output of each algorithm is the connection network of weather stations. The states associated with each node a set of values a particular variable nodes that are capable of being recorded. In this paper for each node four modes has considered. (0=dry, 1=light rain, 2=moderate rain, 4=heavy rain). using Matlab software algorithms discussed were implemented separately. It should be noted that in all algorithms, the number of potential parents or neighbors, 4 nodes is assumed (See the contents stated in the previous section). For example, the K2 algorithm output is shown in Figure 9. The comparing results of all algorithms, with respect to both accuracy and construction time factors implemented in figuers 10,11.

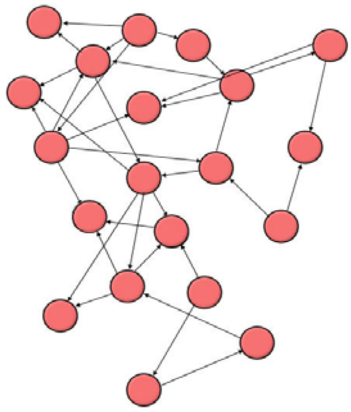
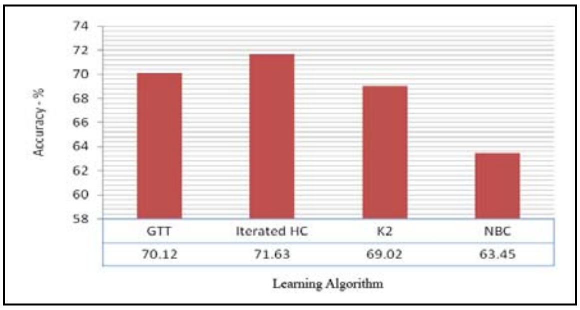


Fig. 9: K2 Algorithm Output For 22 Weather Stations In Area Of Study



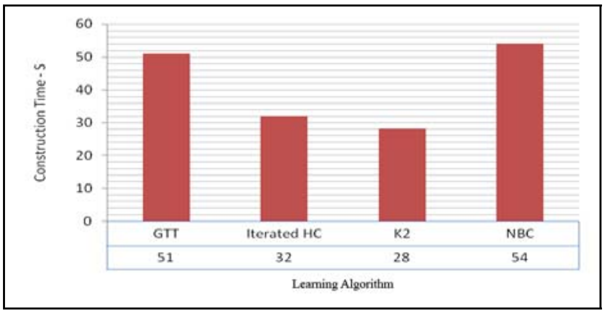


Fig. 10: Comparative Study for Effect of Learning Algorithms On Construction Times With Weather Database

Fig. 11: Comparative Study for Effect of Learning Algorithms On Predictive Accuracy With Weather Database

7. Conclusion

In this paper, Some learning algorithms including Naïve Bayes Classifier, K2, Hill Climbing, Iterative Hill climbing and Greedy Tick Thinning were implemented and compared. K2 Learning Algorithm in this paper demonstrate good performances in finding appropriate structure, and present a relatively low time complexity. As a result, it was shown that from the point of prediction acurracy, the Iterative Hill climbing algorithm is the best algorithm compared to other algorithms, and from the point of construction times, the K2 algorithm is the best algorithm and NBC is worst algorithm. Based on the results obtained for meteorological dataset, we believe that the K2 algorithm can be used more often for bayesian networks structural learning in meteorological applications. As future works, we are interested to investigate the use of evolutionary based algorithms for searching the state space in structural learning.

8. References

[1] Cano Rafael , Sordo Carmen, M. Guti´errez Jos´e, " *Applications of Bayesian Networks in Meteorology Advances in Bayesian Networks*", Springer, March 2004, pp. 309-327.

[2] Hanh Le, Mark Tadross, Anet Potgieter, " Weather Forecasting with Bayesian Networks", *Department Of Computer Science University Of Cape Town*, 2008.

[3] Marco Scutari, "Learning Bayesian Networks with the bnlearn R Package" *Journal of Statistical Software*, Volume 2, Issue 1, 10 Jul 2010.

[4] A. R. Khanteymoori, M. M. Homayounpour, M. B. Menhaj, " *A Bayesian Network Based Approach for Data Classification Using*", 2009.

[5] Evelina Lamma, Fabrizio Riguzzi and Sergio Storari, " Improving the K2 Algorithm Using Association Rule Parameters", *Elsevier*, 2005.

[6] de Campos, L. and Castellano, J. "Bayesian Network Learning Algorithms Using Structural Restrictions", *International Journal of Approximate Reasoning*, 2007, pp. 233-254.

[7] Byoungkoo, L. and Joseph, J. "Learning a Probabilistic model of rainfall using graphical models". *School of Computer Science. Carnegie-Mellon University*, 2006.