The Research on Search Algorithms in the Machine Learning

Hui Liu[1],Yonghui Cao[2]
[1] School of Computer and Information Technology, Henan Normal University, Xin Xiang, 453007, China

[2] School of Economics & Management, Henan Institute of Science and Technology, Xin Xiang, 453003 ,China

Abstract
Machine learning is the estimation of the topology (links) of the network, it can be achieved by utilizing a search algorithm through the possible network structures, because it is finding the best network that fits the available data and is optimally complex. In this paper, a greater importance is given to the search algorithm because we have assumed that the data will be complete. We focus on Two search algorithms are introduced to learn the structure of a Bayesian network in the paper. The heuristic search algorithm is simple and explores a limited number of network structures. On the other hand, the exhaustive search algorithm is complex and explores many possible network structures.

Keywords: *Structural learning, Search Algorithms, Heuristic Search, Exhaustive Search*

1. Introduction

A Bayesian network, Bayes network, belief network or directed acyclic graphical model is a probabilistic graphical model that represents a set of random variables and their conditional dependencies via a directed acyclic graph (DAG). For example, a Bayesian network could represent the probabilistic relationships between diseases and symptoms. Given symptoms, the network can be used to compute the probabilities of the presence of various diseases.

Formally, Bayesian networks are directed acyclic graphs whose nodes represent random variables in the Bayesian sense: they may be observable quantities, latent variables, unknown parameters or hypotheses. Edges represent conditional dependencies; nodes which are not connected represent variables which are conditionally independent of each other. Each node is associated with a probability function that takes as input a particular set of values for the node's parent variables and gives the probability of the variable represented by the node. For example, if the parents are m Boolean variables then the probability function could be represented by a table of 2m entries, one entry for each of the 2m possible combinations of its parents being true or false.

Efficient algorithms exist that perform inference and learning in Bayesian networks. Bayesian networks that model sequences of variables (e.g. speech signals or protein sequences) are called dynamic Bayesian networks. Generalizations of Bayesian networks that can represent and solve decision problems under uncertainty are called influence diagrams.

Because a Bayesian network is a complete model for the variables and their relationships, it can be used to answer probabilistic queries about them. For example, the network can be used to find out updated knowledge of the state of a subset of variables when other variables (the evidence variables) are observed. This process of computing the posterior distribution of variables given evidence is called probabilistic inference. The posterior gives a universal sufficient statistic for detection applications, when one wants to choose values for the variable subset which minimize some expected loss function, for instance the probability of decision error. A Bayesian network can thus be considered a mechanism for automatically applying Bayes' theorem to complex problems.

A Bayesian network is not allowed to have a cycle because of the computational difficulties. A cycle in a Bayesian network leads to a "circular reasoning" between the variables. For example, if the dependencies in above network are: $X_1 \to X_2$, $X_2 \to X_3$, and $X_2 \to X_3$, a cycle will be formed. If evidence is entered into the variable X_1, the Bayesian network will run the evidence to X_2, then to X_3. Then, the evidence will travel to X_1 because X_1 depends on X_3. The evidence may run in the network forever because all the variables depend on each other in a circular way.

A heuristic arc addition is employed not to have a cycle in the Bayesian network while generating the Bayesian structure. An exhaustive arc addition is also employed to explore more network possibilities without limitation. In

the exhaustive arc addition algorithm, a cycle check is employed before and arc is added. The following section presents the details of heuristic and exhaustive search algorithms.

2. Heuristic Search

In the heuristic search algorithm, the variables of the system have to be ordered in a certain way to prevent cycles from being created. The decision variables should be in the last columns in the database; and, the first columns of the database should be filled with the variables without parents, independent variables. After placing the independent variables in the first columns, the children of the independent variables should be placed in the following columns. The rest of the columns are filled with the children of the previously placed variables. Ordering of the variables is necessary because the heuristic arc addition adds the arcs from the first variables to the last variables. Because of the ordering, we need to have some knowledge about the variables. This does not mean that we need to know the dependencies between the variables. For example, let B be a Bayesian network with three variables, $\{X_1, X_2, X_3\}$.If we know the variable X_1 is the first variable and the variable X_2 is the decision node. Then the column order will be $\{X_1, X_2, X_3\}$.

The heuristic search starts with adding and removing arcs from the each variable to the last variable. Let the network have n variables. After adding an arc, the algorithm calculates the network score, records the score in a list, and removes the arc. The algorithm finds the arc that gives the highest increase in the network score. Let us assume that the arc from the kth variable to the last variable, n, gives the highest increase in network score. Then, the algorithm adds the arc from the kth variable to the last variable. After the arc is added, the algorithm adds and removes arcs from the remaining variables to the last variable. Then, the algorithm chooses the arc with the highest score increase and adds the arc to the network. This continues until no increase in the network score can be obtained by adding an arc to the last variable. Then, the algorithm starts adding arcs from the variables $\{1, 2, \cdots n-2\}$ to the $(n-1)th$ node. The algorithm adds arcs to $(n-1)th$ node until there is no increase in the network score. The algorithm stops when it adds an arc from the first variable to the second variable. The following is the heuristic search algorithm used in this research.

(1)Collect data
(2)Define the variables from the available data
(3)Start with a network with no arc.
(4)Estimate the parameters (only independent probabilities) of the BN using the MLE method using initial data
(5)Add a new arc from the ith variable to the jth variable to generate a network candidate and remove the arc. Repeat the process with $i = \{1, 2, \cdots j-1\}$ and generate networks $(B_1, B_2, \cdots, B_{j-1})$. Start j from n and decrease j by 1.
(6)Calculate the scores of the candidate networks and record them in a list.
(7)Find the network (B) with the maximum score and keep it for the next step.
(8)Repeat the steps 5, 6, and 7 until there is no increase in the network score.
(9)If $j > 1$, then go to step 5.
(10)Update the network parameters along with new data
(11)Update the network structure:

If enough new data obtained, go to step 1 and generate a new network structure.

If no structural update is necessary go to step 10.

Consequently, the heuristic search algorithm adds arcs only in the forward direction because this protects the network from having cycles and complex network structure. On the other hand, there is a price of arranging the variables at the creation of the database in the heuristic algorithm. Since the agents will not have much knowledge about the environmental variables, it is hard to arrange the variables at the beginning. There is a need for a better search algorithm that explores more possibilities in the network. The following paragraph introduces another searching algorithm that eliminates the arranging the variables, namely exhaustive search.

3. Exhaustive Search

The exhaustive search algorithm explores all the possible arcs in the network during its execution. The algorithm starts adding arcs from the ith variable to the jth variable where $i = \{1, 2, \cdots, n\}$, $j = \{1, 2, \cdots, n\}$, $i \neq j$. This covers $n \cdot (n-1)$ arcs throughout the network. The algorithm

calculates the network score for each arc addition. Then, it chooses the arc with the highest increase in the network score. The algorithm repeats the above steps until there is no increase in the network score.

There are two major drawbacks in the exhaustive search algorithm. First, the number of arcs to be tried might become intractable when the number of variables is large. Second, during the search, the algorithm might introduce cycles to the network because it can add an arc in any direction. An additional algorithm is incorporated to the search algorithm to keep track of cycles. Using the additional algorithm, the search algorithm checks whether the new arc introduces a cycle or not. If the arc introduces a cycle, the algorithm does not add the arc to the network. The following is the exhaustive search algorithm used in this research.

(1)Collect data
(2)Define the variables from the available data
(3)Start with an empty network
(4)Estimate the parameters (only independent probabilities) of the BN using the MLE method using initial data
(5)Add a new arc from the ith variable to the jth variable to create a candidate network and remove the arc. Repeat the process for every value of i and j where $i=\{1,2,\cdots,n\}$, $j=\{1,2,\cdots,n\}$, $i\neq j$ .This step creates m possible networks $(B_1,B_2,\cdots,B_m)$. Algorithm creates $m=n\cdot(n-1)$ networks in first visit to step 5.
(6)Remove the network with cycles from the candidate fist.
(7)Calculate the scores of the candidate networks and record it in a list.
(8)Find the network (B) with the maximum score and keep it for the next step.
(9)Do step 5 through 8 until there is no increase in the network score.
(10)Update the network parameters along with new data
(11)Update the network structure:
If enough new data obtained, go to step 1 and generate a new network structure.
If no structural update is necessary go to step 10.
The search algorithms are explained in detail. There is a need to analyze the complexity of the search algorithm before there are implemented. The following section gives the complexity analysis of both search algorithms.

4. Complexity Analysis for Search Algorithms

As stated earlier, the heuristic search algorithm needs prior knowledge about the variables in terms of their order in the database. On the other hand, the number of iterations in the heuristic search algorithm may be tractable. In the heuristic search, the algorithm tries $(n-1)$ arcs in the first trip from step 5 to step 7. The algorithm repeats steps 5 through 7 until there is no increase in the network score. Assuming the algorithm adds an arc in every trip, the number of arcs tried will be one less then the previous trip. Algorithm can repeat step 5 through 7 at most $(n-1)$ times. In $(n-1)$ trips, the algorithm generates $(n-1)+(n-2)+\cdots+1$ networks candidates. When the algorithm reaches step 8, the algorithm loops back to step 5 and repeats the same process for the variables $\{X_{n-1},X_{n-2},\cdots,X_2\}$ .Therefore, after the first loop, the algorithm generates $(n-1)+(n-2)+\cdots+1$ network candidates. The complexity of the heuristic search algorithm is denoted as C_h.

In the following complexity analysis, each loop shows the number of network candidates tried until the algorithm reaches to the step 8. Since the algorithm will repeat itself for $(n-1)$ variables, the analysis has $(n-1)$ loops as the following.

Loop1

$$(n-1)+(n-2)+\cdots+1=n(n-1)-(1+2+\cdots+(n-1))$$
$$=n(n-1)-\frac{n(n-1)}{2}=\frac{n(n-1)}{2}$$

Loop2

$$(n-2)+(n-3)+\cdots+1=\frac{(n-1)(n-2)}{2}$$

$\vdots$

$\vdots$

Loop (n-1) $$\frac{(n-(n-1))(n-(n-2))}{2}=1$$

If we add the number of candidate networks from each loop, the following can be obtained:

$$C_n=\frac{n(n-1)+(n-1)(n-2)+\cdots+(n-(n-1))(n-(n-2))}{2}$$
$$C_n=\frac{2(n-1)^2+2(n-3)^2+\cdots+2(n-(n-2))^2}{2}$$

Then, we can further modify the equation as follows:

$$C_n = (n-1)^2 + (n-3)^2 + \cdots + 2(n-(n-2))^2 \quad (1)$$

Since each element in C_n is less than n^2. We can state that

$$C_n < n^2(n-3) < n^3 \quad (2)$$

Equation (2) illustrates the complexity of the heuristic search. The following paragraphs will explore the complexity of the exhaustive search algorithm.

The exhaustive search algorithm tries every possible arc in the network during its first visit to step 5. In a graph with n nodes, there can be $n(n-1)$ possible directed edges. Therefore, the algorithm generates $n(n-1)$ network candidates and the complexity of the first visit is $n(n-1)$.Then the algorithm continues until it reaches to step 9 and loops back to step 5 until there is no increase in the network score.

After the first loop, the complexity decreases by 1 in each step because the algorithm will not try the arc added in the previous step. The following presents the complexity analysis of the exhaustive search algorithm. First, the complexity is calculated for each loop. Then, they are added to obtain the complexity of the algorithm.

Loop 1 $n(n-1)$

Loop 2 $n(n-1)-1$

$\vdots$

$\vdots$

Loop N $n(n-1)-N+1$

The exhaustive search algorithm does not perform a certain number of loops. The algorithm will continue until there is no increase in the network score. Therefore, we will assume that the algorithm end after N loops for the complexity calculations. If we add the complexities of all the loops together, the complexity of the exhaustive search, C_e, becomes the following.

$$C_e = n(n-1)N - (1+2+\cdots+(N-1)) \quad (3)$$

$$C_e = n(n-1)N - \frac{N(N-1)}{2} \quad (4)$$

If the network has great number of arcs, then the complexity of the algorithm becomes large. For example, if the algorithm ends in step $N=n$, the complexity becomes

$$C_e = n^2(n-1) - \frac{n(n-1)}{2} = \frac{2n^2(n-1) - n(n-1)}{2} \quad (5)$$

$$C_e = \frac{(n-1)n(2n-1)}{2} \quad \text{for } n=N \quad (6)$$

In general, number of nodes in a Bayesian network, n is much larger than 1. Therefore, we can reevaluate the complexity by assuming $n \rangle\rangle 1$ .The following equation represents the computational complexity of the exhaustive search algorithm when the number of steps is equal to the number of variables.

$$C_e \cong \frac{n \cdot n \cdot 2n}{2} = \frac{2n^3}{2} \Rightarrow C_e \cong n^3 \quad (7)$$

As can be seen above, the complexity of the exhaustive algorithm is larger than the complexity of the heuristic algorithm when $N=n$.

For the networks with large number of variables (nodes), the algorithm does not stop when $N=n$.Let us calculate the worst case scenario for the exhaustive algorithm. The algorithm might explore all possible arcs in the network, which is equal to $n(n-1)$.This is true because a complete graph with n nodes has $n(n-1)$ possible directed edges [30]. Therefore, we will replace N with $n(n-1)$ in the complexity analysis. Then, the complexity of the exhaustive search algorithm becomes the following.

$$C_e = n(n-1)N - \frac{N(N-1)}{2} = n(n-1)n(n-1) - \frac{n(n-1)(n(n-1)-1)}{2} \quad (8)$$

$$C_e = \frac{2n^2(n-1)^2 - n^2(n-1)^2 - n(n-1)}{2} = \frac{n^2(n-1)^2 - n(n-1)}{2} \quad (9)$$

We can simplify the equation above by assuming $n >> 1$. In this case, the complexity of the algorithm becomes the following.

$$C_e \cong \frac{n^2 \cdot n^2 - n^2}{2} = \frac{n^2(n-1)^2}{2} \Rightarrow C_e \cong \frac{n^4}{2} \quad (10)$$

5. CONCLUSION

Bayesian network is a complete model for the variables and their relationships.Two search algorithms are introduced to learn the structure of a Bayesian network.
The heuristic search algorithm is simple and explores a limited number of network structures. the heuristic search algorithm adds arcs only in the forward direction because this protects the network from having cycles and complex network structure. there is a price of arranging the variables at the creation of the database in the heuristic algorithm. Since the agents will not have much knowledge about the environmental variables. There is a need for a better search algorithm that explores more possibilities in the network.

The exhaustive search algorithm is complex and explores many possible network structures. The heuristic search algorithm needs prior knowledge about the variables in terms of their order in the database. If the network has great number of arcs, then the complexity of the algorithm becomes large. The complexity of the exhaustive algorithm is approximately n-fold larger than the complexity of the heuristic search algorithm.

Acknowledgments

This work is financially supported by Science and Technology research projects by the Education Department of Henan province (Project No. 12A520025). Thanks for the help.

References

[1]F.V. Jensen, an Introduction to Bayesian Networks. London, UK: University College London Press, 2012.
[2]Ali Jarrahi and Mohammad Reza Kangavari, "An Architecture for Context-Aware Knowledge Flow Management Systems," International Journal of Computer Science Issues, vol. 9,2012.
[3]Y.Shoham, "Agent-oriented programming," Artificial intelligence, vol. 60(1), pp. 51-92, 2012 .
[4]J.Pearl, "Bayesian networks", in M. Arbib (Ed.), Handbook of Brain Theory and Neural Networks, MTT Press, pp. 149-153, 2012
[5]J.Pearl, "Bayesian networks," Technical Report R-246, MTT Encyclopedia of the Cognitive Science, October ,2011.
[6]F.V. Jensen, "Bayesian network basics," AISB Quarterly, vol. 94, pp. 9-22, 2011.
[7]W.Lam and F. Bacchus, "Learning Bayesian belief networks: an approach based on the MDL principle," Computational Intelligence, vol. 10, pp. 269-293, 2011.
[8]N.Friedman, M. Goldszmidt, D. Heckerman, and S. Russell, "Challenge: Where is the impact of the Bayesian networks in learning?" In Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI), pp.10-15, 2011.
[9]N.Friedman, K. Murphy, and S. Russell, "Learning the structure of dynamic probabilistic networks," in G.F. Cooper and S. Moral (Eds.), Proceedings of Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI '98), San Francisco, CA: Morgan Kaufmann, 2010.
[10]Amir Mosavi, "Multiple Criteria Decision-Making Preprocessing Using Data Mining Tools," International Journal of Computer Science Issues, vol. 7,2010.
[11]B.Theisson, C. Meek, and D. M. Chickering, and D. Heckerman, "Learning mixtures of Bayesian networks," in G.F. Cooper and S. Moral (Eds.), Proceedings of Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI '98), San Francisco, CA: Morgan Kaufmann, 2010.
[12]N.Friedman, "The Bayesian structural EM algorithm," in G.F. Cooper and S. Moral (Eds.), Proceedings of Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI '98), San Francisco, CA: Morgan Kaufmann, 2010.
[13]D.Spiegelhalter, P. Dawid, S. L. Lauritzen, and R. Cowell, "Bayesian analysis in expert systems," Statistical Science, vol. 8, pp. 219-282, 2009.
[14]D.Heckerman, D. Gieger, and M. Chickering, "Learning Bayesian networks: The combination of knowledge and statistical data," Technical Report MSR-TR-94-09, Microsoft Research, Redmond, WA, 2009.
[15]C.Claus, "Dynamics of multi-agent reinforcement learning in Cooperative multi-agent systems," Ph.D. Dissertation, Univ. of British Colombia, Canada, 2009.
[16]S. Sen and M. Sekaran, "Multi-agent coordination with learning classifier systems," in Proceedings of the IJCAI Workshop on Adaptation and Learning in Multi-agent Systems, Montreal, pp. 84-89, 2009.
[17]C.Boutilier, "Planning, learning and coordination in multi-agent decision processes," in Sixth conference on Theoretical Aspects of Rationality and Knowledge (TARK'96), The Netherlands, 2008.

First Author Hui Liu received the MS degree in computer education from Henan Normal University in 2008.She is currently a teacher in Henan Normal University. Her research interest is in the areas of machine learning.

Second Author Yonghui Cao received the MS degree in business management from Zhejinag University in 2006. He is currently a doctorate candidate in Zhejiang University. His research interest is in the areas of machine learning.