ISSN:0975-9646

Siva Rama Krishna Prasad Jaladi et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (5) , 2012,5227-5230

Bayesian Network and Variable Elimination Algorithm for Reasoning under Uncertainty

Siva Rama Krishna Prasad Jaladi , Dharmaiah Devarapalli

Department of Computer Science and Engineering,
Vignan's Institute of Technology, JNTU Kakinada Beside VSEZ,
Duvvada, Visakhapatnam,
Andhra Pradesh, India

Abstract - A common task for a Bayesian network is to perform inference by computing to determine various probabilities of interest from the model. We are using an algorithm for construction of Bayesian network from given data input from several data sources such as Oracle, Access, Excel, etc., and variable elimination algorithm for answering probabilistic queries with respect to a Bayesian network. Our algorithm makes use of XML Bayesian Interchange Format to support portability of constructed network within modules of program. The algorithm runs in time and space exponential in the tree width of the network. The variable elimination algorithm acts on a set of factors. Each factor involves a set of variables and each node in a Bayesian network is equipped with a conditional probability function that expresses the likelihood that the node will take on different values given the values of its parents. The initial sets of factors are the network's conditional probability distributions (tables). The probability distributions constructed during variable elimination in Bayesian networks have always been denoted as probability.

Index Terms - Bayesian network construction; Inference; XML BIF; Conditional probability query; Decision making

INTRODUCTION

Bayesian networks are graphical models whose nodes represent random variables and whose edges represent conditional dependence between variables. A Bayesian network consists of a directed acyclic graph and a corresponding set of CPTs (Conditional Probability Tables). Based on the conditional independencies holding in the directed acyclic graph, the product of the CPTs is a discrete joint probability distribution.

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.

BAYES' THEOREM

Given two events E and F such that P (E)≠ 0 and P (F) ≠ 0, then Bayes theorem is

$$P(E|F) = \frac{P(F|E)P(E)}{P(F)}$$

Given n mutually exclusive and exhaustive events E1, E2… En such that P (Ei) ≠ 0 for all i, and for $1 \le i \le n$,

$$P(Ei \mid F) = (P(F \mid Ei)P(Ei))/(P(F \mid E1)P(E1) + P(F \mid E2)P(E2) + \cdots + P(F \mid En)P(En))$$

Let X and Y be random variables with possible values x and y, Then the joint probability distribution of X and Y is P(X = x, Y = y)

Let X and Y be discrete random variables with possible values x and y, and a joint probability distribution P(X = x, Y = y). Then

$$P(X = x) = \sum_{y} P(X = x, Y = y)$$

Let A, B, and C is sets of random variables defined on the same probability space. Then sets A and B are said to be conditionally independent given the set C if, for all values of the variables in the sets a, b, and c, whenever P(c)≠0, the events A = a and B = b are conditionally independent given the event C = c. That is, either P(a|c) = 0, or P(b|c) =0 , or P(a|b,c) = P(a|c)

MUTUAL INFORMATION

Mutual information is nothing but information flow between two nodes. In information theory, the mutual information of two nodes Xi, Xj is defined as

$$I(X_i, X_j) = \sum_{x_i, x_j} P(x_i, x_j) \log \frac{P(x_i, x_j)}{P(x_i)P(x_j)}$$

and conditional mutual information is defined as

$$I(X_i, X_j|C) = \sum_{x_i, x_j, c} P(x_i, x_j, c) \log \frac{P(x_i, x_j|c)}{P(x_i|c)P(x_j|c)}$$

This equation can be modified as

$$I(X_i, X_j) = \sum P(x_i) p(x_j|x_i) \log (p(x_j / x_i)/p(x_j))$$

Where Xi and X j are two nodes and C is a set of nodes

BAYESIAN INFERENCE

Inference on Bayesian network computes the marginal probability P (V = v) for each node V and each possible instantiation v. Inference gives an idea of how likely cases for a specific random variable by using the information in the Bayesian network. Inference can also be done on a Bayesian network when the values of some nodes (evidence) are known and wish to compute the likelihood of values of other nodes. This is computation of posterior probabilities, since it finds P (V = v|e) for each node V and each possible instantiation v, given the evidence value e.

Graphical models allow the representation of large domain probabilistic models by encoding their qualitative properties. Several methods for exact and appropriate inference to suit different kinds of graphical models have been developed. This paper deals with only exact inference after construction of Bayesian belief network from given data source.

EXACT INFERENCE

The inference refers to the task of calculating the likelihood of some evidence e, P (e), as well as the task of calculating the marginal distribution (P (V = v)) or posterior probabilities for given initiation variable v, (P (V = v|e)) for each instantiation v of each node V, possible given some evidence e of other node values.

ARCHITECTURE

The proposed implementation consists of two modules – Bayesian network construction algorithm and Inference algorithm. Following diagram shows the architecture of the system.

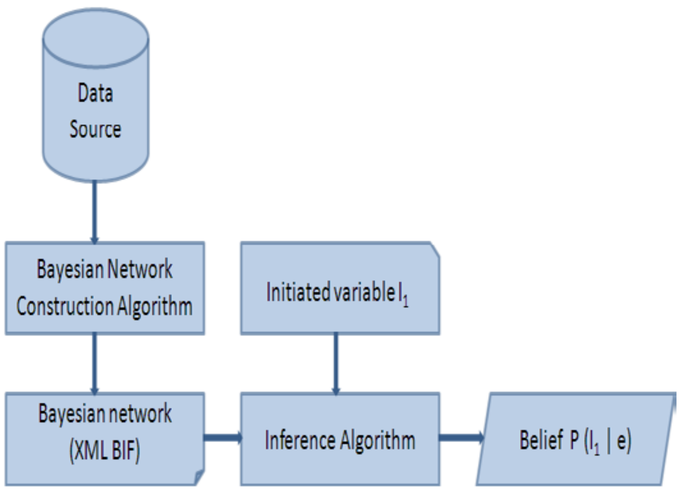

Figure 1: Architecture Diagram

Bayesian network construction algorithm is capable of accepting given input from different data sources such as MS Access, MS Excel, Oracle, etc., and computes mutual information among variables and prepares nodes and conditional probability table. Once the algorithm finishes, the result can be saved as XML Bayesian Interchange Format.

Bayesian Inference algorithm accepts Bayesian network data generated in Bayesian network construction algorithm in the form of XML Bayesian Interchange Format, initiation variable I1 and computes the posterior belief P (I1 | e) for any given ordering of the variables which is initiated by I1

BAYESIAN NETWORK CONSTRUCTION ALGORITHM DESCRIPTION

A Bayesian belief network for reliability prediction and management was constructed using this algorithm [8]. The validation of model's problem structure and behavior is a continuous process till the end of the construction process is completed. This algorithm has three phases: ***drafting, thickening and thinning.***

In the first phase, this algorithm computes mutual information of each pair of nodes as a measure of closeness, and creates a draft based on this information. In the second phase, the algorithm adds arcs when the pairs of nodes cannot be d-separated. The result of Phase II is an independence map (I-map) of the underlying dependency model. In the third phase, each arc of the I-map is examined using CI tests and will be removed if the two nodes of the arc can be d-separated. The result of Phase III is the minimal I-map.

THE ALGORITHM

Input Ms-Access, Excel and text file formats data

Output Generates belief network and can be exported to Bayesian Interchange Format

PHASE I: (DRAFTING)

1. Initiate a graph G (V, E) where V= {all the nodes of a data set}, E= { }. Initiate two empty ordered set S, R.
2. For each pair of nodes (v_i,v_j) where v_i,v_j ε V, compute mutual information using equation. For the pairs of nodes that have mutual information greater than a certain small value e, sort them by their mutual information from large to small and put them into an ordered set S.
3. Get the first two pairs of nodes in S and remove them from S. Add the corresponding arcs to E.
4. Get the first pair of nodes remained in S and removes it from S. If there is no open path between the two nodes, add the corresponding arc to E; otherwise, add the pair of nodes to the end of an ordered set R.
5. Repeat step 4 until S is empty.

PHASE II: (THICKENING)

1. Get the first pair of nodes in R and remove it from R.
2. Find a block set that blocks each open path between these two nodes by a set of minimum number of nodes.(This procedure identify_block_set (current graph, node1, node2) is given at the end of this subsection.)
3. Conduct a CI test. If these two nodes are still dependent on each other given the block set, connect them by an arc.
4. Go to step 1 until R is empty.

PHASE III: (THINNING)

1. For each arc in E, if there are open paths between the two nodes besides this arc remove this arc from E temporarily and call procedure identify_block_set (current graph, node1, node2).
2. Conduct a CI test on the condition of the block set.

If the two nodes are dependent, add this arc back to E; otherwise remove the arc permanently.

INFERENCE ALGORITHM DESCRIPTION

Inference in BBNs represents the stage in which the networks are used for supporting decision making. Scenario analysis through BBNs may be used for decision support in generic way as follows

1. The outcome (value of the leaf/child node) can be calculated as a result of different inputs to the model (values of the parent/root nodes)
2. The likelihood of the input (values of the parent/root nodes) that corresponds to a certain outcome (value of the child/leaf node) can be calculated
3. Both ways of decision support can be combined.

A situation can be characterized in terms of the inputs (pre-defined values of input variables). In the other direction, the most likely values of inputs that correspond to particular outcomes can be calculated.

THE ALGORITHM

Input A belief network BN = {P1,…, Pn} an ordering of the variables d, Initiated variable e

Output Belief P (I_1 | e) network for all possible queries

I. INITIALIZE

1. Generate an ordered partition of the conditional probability matrices, $P_1 \ldots P_n$, where P_i contains all matrices whose highest variable is Xi.
2. Put each observed variable in its bucket.
3. Let $S_1 \ldots S_j$ be the subset of variables in the processed bucket on which matrices are defined.

II. FORWARD PASS

The most probable value is obtained by the product in P1. Most probable explanation is obtained by assigning values in the ordering d consulting recorded functions in each P as follows

1. Given the assignment $x = (x_1, \ldots, x_{i-1})$ choose $x_i = x_i^0(x)(x_i^0 \text{ is in } P_i)$ or
2. Choose $x_i = argmax_{x_i} \prod_{\{h_j \notin p_i | x=(x_1,\ldots x_{i-1})\}} h_j$

III. BACKWARD PASS

1. For p = n down to 1
2. Do
3. for all the matrices $h_1, h_2, \ldots, h_j$ in P_p, do
4. If (observed variable) P_p contains $X_p = x_p$, assign $X_p = x_p$ to each h_i and put each in appropriate bucket
5. else,

$$\cup_p = \cup_{i=1}^{j} S_i - \{X_p\}$$

6. Generate functions

$$h_p = max_{x_p} \prod_{i=1}^{j} h_i \ and$$

$$X_p^0 = argmax_{x_p} h_p$$

7. Add hp to bucket of largest-index variable in $\cup_p$.
8. Done

EXISTING METHODS

Application of BBNs on a larger scale has been strongly dependent on the availability of efficient algorithms and the availability of computing power. For example, in problems concerning with monitoring and detection systems such as credit card fraud detection [1] [9], energy theft detection, medical diagnostic systems, Email spam filters, etc., we need to know the probability of given observations of the other variables. This probability is not stored directly in the model, and hence needs to be computed. The computation of a probability of interest given a model is known as probabilistic inference [3].

Two common exact inference algorithms available are – Pearl's algorithm [2] [10] [11] and the Lauritzen - Spiegelhalter algorithm [2] [10] [11]. While all existing exact inference algorithms for general Bayesian networks do have exponential running times, inference can be performed in linear time on a polytree using Pearl's algorithm [2][10][11]. Thus if a network could be approximated by a polytree, inference could be performed very quickly and this will not work for all the cases

We are using Variable Elimination [7] algorithm as it generalizes dynamic programming to accommodate algorithms for many complex problem solving and reasoning activities, including adaptive consistency for constraint satisfaction, Fourier and Gaussian elimination for linear equalities and inequalities, and dynamic programming for combinatorial optimization.

RESULTS AND DISCUSSION

In this paper, we combined Bayesian network construction using three phases Bayesian network construction algorithm and then used another algorithm for variable elimination, compilation, and structured representations of factors. For a directed graphical model, we can use this factored representation of the JPD to do marginalization efficiently. The key idea is to push sums as far as possible when marginalizing out irrelevant terms. The principle of distributing sums over products can be generalized greatly to apply to any type of decision making problems. The amount of work we perform when computing a marginal is bounded by the size of the largest term that is encountered. The following figures shows the output of Bayesian network constructed by using three phase algorithm and resultant Bayesian network after applying variable elimination algorithm.

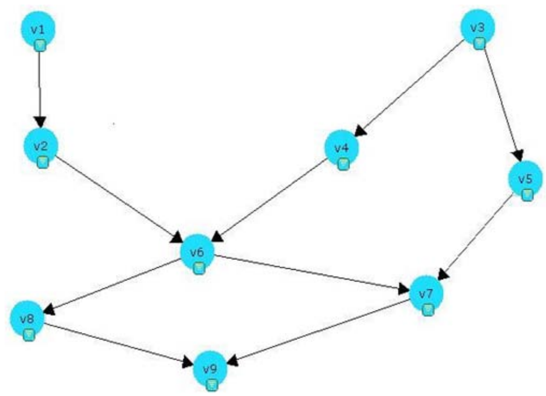

Figure 2: Input Bayesian network (DAG) with 9 nodes

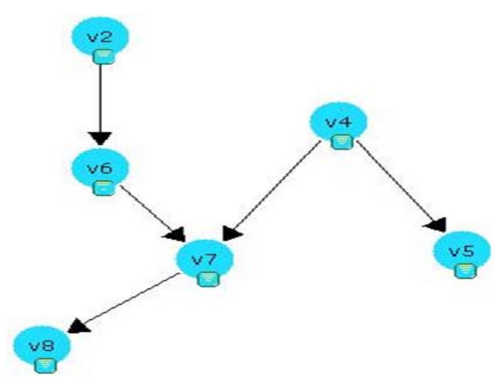

Figure 3: Result Bayesian Net for inferring the likehood

The complexity of the Variable elimination algorithm depends on time and space required to process and it is time and space is exponential in number of variables. We chose this variable elimination algorithm to minimize this to NP-hard.

CONCLUSION

The basic task for probabilistic inference system is to compute the posterior probability distribution for a set of query variables, given some observed event. During the algorithm, two factor operations are performed many times: factors are multiplied and a variable is summed out of a factor. These factor operations reduce the task of performing many multiplication and addition operations on real–numbers. Given a function f (X) in factored form, and some corresponding query, the method will eliminate a variable X from this function to produce another function f (X− Xi), while ensuring that the new function is as good as the old function as far as answering the query of interest. The idea is then to keep eliminating variables one at a time, until we can extract the answer we want from the result. The key idea is that when eliminating a variable, we will only need to multiply factors that mention the eliminated variable.

REFERENCES

[1] Efstathios Kirkos, Charalambos Spathis, Yannis Manolopoulos, Data Mining techniques for the detection of fraudulent financial statements , Elsevier Expert systems with applications 32 (2007) 995 - 1003

[2] Neapolitan, R.E. , Probabilistic reasoning in expert systems: theory and algorithms, John Wiley & Sons, 1990.

[3] Haipeng Guo, William Hsu, A Survey of Algorithms for Real-Time Bayesian Network Inference, Laboratory for Knowledge Discovery in Databases

[4] RONANDALY, QIANG SHEN and STUART AITKEN, Learning Bayesian networks: approaches and issues, The Knowledge Engineering Review, Vol. 26:2, 99–157. Cambridge University Press, 2011

[5] Stuart Russell, Peter Norvig , Artificial Intelligence, A modern approach Third Edition, PEARSON publications

[6] James D. Park,Adnan Darwiche, Complexity Results and Approximation Strategies for MAP Explanations , Journal of Artificial Intelligence Research 21 (2004) 101-133

[7] F. van Harmelen, V. Lifschitz and B. Porter, Handbook of Knowledge Representation, Elsevier B.V., 2008

[8] Jie Cheng, David A. Bell, Weiru Liu , Learning Belief Networks from Data: An Information Theory Based Approach

[9] Rose F. Liu, Rusmin Soetjipto, Analysis of Three Bayesian Network Inference Algorithms: Variable Elimination, Likelihood Weighting, and Gibbs Sampling

[10] David Heckerman, A Tutorial on Learning With Bayesian Networks, Technical Report, Microsoft Research, March 1995 (Revised November 1996)

[11] V.Dheepa , Dr. R.Dhanapal , A short paper on Analysis of Credit Card Fraud Detection Methods, International Journal of Recent Trends in Engineering, Vol 2, No. 3, November 2009

[12] Pearl, J., 2000. Causality: models, reasoning and inference. Cambridge University Press

[13] Denver Dash, Marek J. Druzdzel, Robust Independence Testing for Constraint-Based Learning of Causal Structure, In Proceedings of the Nineteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-03), pages 167-174, Morgan Kaufmann Publishers, Inc., San Francisco, CA, 2003

[14] Pearl, J. : Probabilistic reasoning in intelligent systems – Networks of plausible inference, Morgan – Kaufmann 1988