International Open Journal of Operations Research
Vol. 1, No. 2, May 2013, PP: 16 – 39, ISSN: 2328-8582 (ONLINE)
Available online at http://www.acascipub.com/Journals.php

Research article

Significant Role and Special Aspects of Inference Mechanism in Bayesian Network

Prof. (Dr.) V. N. Maurya
Professor & Ex Principal,
Shekhawati Engineering College, Rajasthan Technical University, India
E-mail:prof.drvnmaurya@gmail.com, prof_vnmaurya@yahoo.in

Diwinder Kaur Arora
Inspector of Police,
Group Centre, Central Reserve Police Force, Lucknow-226002, U.P.
Ministry of Home Affairs, Govt. of India
E-mail: hkdkarora@rediffmail.com, diwi.kaur1992@gmail.com

Er. Avadhesh Kumar Maurya
Assistant Professor, Department of Electronics & Communication Engineering
Lucknow Institute of Technology, U.P. Technical University, Lucknow-226002, India
E-mail: avadheshmaurya09@gmail.com

Abstract

Present paper describes some fundamental concepts including Bayesian rule, Bayesian network, impact of Markov condition and Markov equivalence, simple inference and OR-gate inference in the domain of Bayesian network inference (BNI). Bayesian network finds its applications widely in applied statistics, production industries, machine learning, data mining, diagnosis etc. In this paper, our central attention has been made to explore the essence of inference mechanism as a significant domain of Bayesian network (BN) by way of presenting numerical illustrations too. Particularly, significant role and special aspects of inference mechanism in Bayesian network inference have been focused. At the end, some important conclusions are drawn.

Keywords: Bayesian network inference, Bayesian rule, Markov condition and Markov equivalence, OR-Gate inference, Pearl's message propagation algorithm, posterior probability, optimal factoring, serial path, convergent path, divergent path, uncoupled chain etc.

1. Introduction

Bayesian network is applied widely in applied statistics, graph theory, production industries, machine learning, data mining, diagnosis etc. Basically there are three main domains-inference mechanism, parameter learning and structure learning in Bayesian network inference. In this paper, the first domain of inference mechanism has been dealt exhaustively. The inference mechanism articulates the usability of Bayesian network. Bayesian network has a solid evidence-based inference which is familiar to human intuition. However Bayesian network causes a little confusion because there are many complicated concepts, formulas and diagrams relating to it. Such concepts should be organized and presented in clear manner so as to be easy to understand it. A few researchers [1, 2, 3…7] in this direction are worth mentioning. Keeping in view the aforesaid complexity of Bayesian network, the present paper includes two main parts that cover principles of Bayesian network: part 1- Basic concepts and part 2- Bayesian network inference.

International Open Journal of Operations Research
Vol. 1, No. 2, May 2013, PP: 16 – 39, ISSN: 2328-8582 (ONLINE)
Available online at http://www.acascipub.com/Journals.php

2. Basic Concepts

2.1 Bayesian Rule

Bayesian inference, a form of statistical method, is responsible for collecting evidences to change the current belief in given hypothesis. The more evidences are observed, the higher degree of belief in hypothesis is. First, this belief was assigned an initial probability. When evidences were gathered enough, the hypothesis is considered trustworthy.
Bayesian inference was based on Bayesian rule with some special aspects:

$$P(H \mid E) = \frac{P(E \mid H) * P(H)}{P(E)} \tag{2.1.1}$$

where H is probability variable denoting a hypothesis existing before evidence and E is also probability variable notating an observed evidence.
$P(H)$ is *prior probability* of hypothesis and $P(H \mid E)$ which is the conditional probability of H with given E, is called *posterior probability*. It tells us the changed belief in hypothesis when occurring evidence.
$P(E)$ is the probability of occurring evidence E together all mutually exclusive cases of hypothesis. If H and E are discrete, $P(E) = \sum_H P(E \mid H) * P(H)$ otherwise $f(E) = \int f(E \mid H) f(H) dH$ with H and E being continuous, f denoting probability density function.
When $P(E)$ is constant value, $P(E \mid H)$ is the *likelihood function* of H with fixed E. Likelihood function is often used to estimate parameters of probability distribution.

2.2 Bayesian Network

Bayesian network (BN) is the directed acyclic graph (DAG) in which the nodes (vertices) are linked together by directed edges (arcs); each edge expresses the dependence relationships between nodes. If there is the edge from node A to B, we call "A causes B" or "A is parent of B", in other words, B depends conditionally on A. So the edge $A \rightarrow B$ denotes parent-child, prerequisite or cause-effect relationship. Otherwise there is no edge between A and B, it asserts the conditional independence. Let $V=\{X_1, X_2, X_3, \ldots, X_n\}$ and E be a set of nodes and a set of edges, the BN is denoted as below:

$G=(V, E)$ where G is the DAG, V is a set of nodes and E is a set of edges

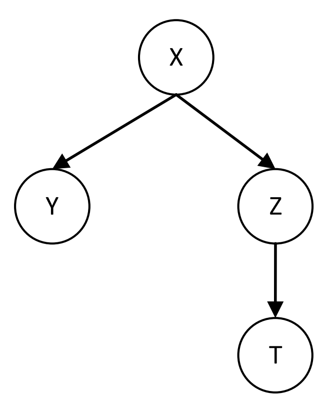


Figure 2.2.1: Bayesian network

Note that node X_i is also random variable. In this paper the uppercase letter (for example X, Y, Z, etc.) denotes random variables or set of random variables; the lowercase letter (for example x, y, z, etc.) denote its instantiation. We should glance over other popular concepts.

- If there is an edge between X and Y ($X \rightarrow Y$ or $X \leftarrow Y$) then X and Y are called *adjacent* each other (or *incident* to the edge).

International Open Journal of Operations Research
Vol. 1, No. 2, May 2013, PP: 16 – 39, ISSN: 2328-8582 (ONLINE)
Available online at http://www.acascipub.com/Journals.php

- Given k nodes $\{X_1, X_2, X_3, \ldots, X_k\}$ in such a way that every pair of node (X_i, X_{i+1}) are incident to the edge $X_i \rightarrow X_{i+1}$ where $1 \leq i \leq k-1$, all edges that connects such k nodes compose a *path* from X_1 to X_k denoted as $[X_1, X_2, X_3, \ldots, X_k]$ or $X_1 \rightarrow X_2 \rightarrow \ldots \rightarrow X_k$. The nodes $X_2, X_3, \ldots, X_{k-1}$ are called *interior* nodes of the path. The *sub-path* $X_m \rightarrow \ldots X_n$ is a path from X_m to X_n: $X_m \rightarrow X_{m+1} \rightarrow \ldots \rightarrow X_n$ where $1 \leq m < n \leq k$. The *directed cycle* is a path from a node to itself. The *simple path* is a path that has no directed cycle. The DAG is the graph that has no directed cycle.
- If there is a path from X to Y then X is called *ancestor* of Y and Y is called *descendant* of X. If Y isn't a descendant of X, Y is called *non-descendent* of X.
- If the direction isn't considered then edge and path are called *link* and *chain*, respectively. Link is denoted $A – B$. Chain is denoted $A – B – C$, for example.

Graph G is a tree if every node except root has only one parent. G is called single-connected if there is only one chain (if exists) between two nodes. Almost BN (s) surveyed here are single-connected DAG (s).

The strength of dependence between two nodes is quantified by conditional probability table (CPT). In continuous case, CPT becomes conditional probability density function (CPD). So each node has its own local CPT. In case that a node has no parent, its CPT degenerates into prior probabilities. For example, suppose X_k is binary node and it has two parents X_i and X_j, the CPT (or CPD) of X_k which is the conditional probability $P(X_k \mid X_i, X_j)$ has eight entries:

$$
\begin{array}{ll}
P(X_k=1|X_i=1, X_j=1) & P(X_k=0|X_i=1, X_j=1) \\
P(X_k=1|X_i=1, X_j=0) & P(X_k=0|X_i=1, X_j=0) \\
P(X_k=1|X_i=0, X_j=1) & P(X_k=0|X_i=0, X_j=1) \\
P(X_k=1|X_i=0, X_j=0) & P(X_k=0|X_i=0, X_j=0)
\end{array}
$$

It is asserted that if X_i is binary node and has n parents then its CPT has 2^{n+1} entries. However only 2^n entries are specified in practice due to $P(X_i=0 \mid \ldots) = 1- P(X_i=1|\ldots)$ when X_i is binary. In case that X_i has k possible values, each CPT has k^n entries.

Example 2.2.1: Suppose event "*cloudy*" is cause of event "*rain*". Events "*rain*" and "*sprinkler*" which in turn is cause of "*grass is wet*". So we have three causal-effect relationships of: *1*-cloudy to rain, *2*- rain to wet grass, *3*-sprinkler to wet grass. This model is expressed below by BN with four nodes and three arcs corresponding to four events and three relationships. Every node has two possible values *True* (*1*) and *False* (*0*) together its CPT.

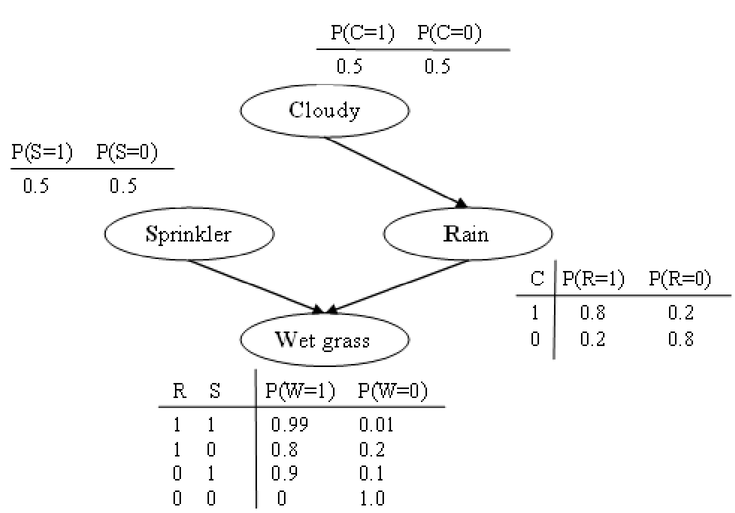


Figure 2.2.2: Bayesian network with CPT (s) in example 2.2.1

Let PA_i be the set of parents of node X_i, the *joint probability distribution* of whole BN is defined as product of CPT(s) or CPD(s) in continuous case of all nodes.

$$P(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} P(X_i \mid PA_i) \qquad (2.2.2)$$

So BN is represented by its joint probability distribution *P* and its DAG. (*G*, *P*) where *G*=(*V*, *E*) is a DAG and *P* is joint probability distribution.
Suppose Ω_i is the subset of PA_i such that X_i must depend conditionally and directly on every variable in Ω_i. In other words, there is always an edge from each node in Ω_i to X_i and no intermediate node between them. This criterion is called as Markov condition which will be discussed later. The joint probability *P* is re-written as below:

$$P(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} P(X_i \mid \Omega_i) \qquad (2.2.3)$$

Back the "wet grass" BN in example 2.2.1, the joint probability is:

$$P(C, R, S, W)=P(C)*P(R)*P(R|C)*P(S|C)*P(W|C,R,S)$$

We have $P(S \mid C) = P(S)$ due to the conditional independence assertion about variables *S* and *C*. Furthermore, because *S* is intermediate node between *C* and *W*, we should remove *C* from $P(W \mid C, R, S)$, hence, $P(W \mid C, R, S) = P(W \mid R, S)$. In short, the joint probability is shown below:

$$P(C, R, S, W)=P(C)*P(S)*P(R|C)*P(W|R,S)$$

2.3 Bayesian Network Inferences

Using Bayesian reference, we need to compute the posterior probability of each hypothesis node in network. In general, the computation based on Bayesian rule is known as the inference in Bayesian network.
Reviewing example 2.2.1, suppose *W* becomes evidence variable which is observed the fact that the grass is wet, so, *W* has value *1*. There is request for answering the question: how to determine which cause (sprinkler or rain) is more possible for wet grass. Hence, we will calculate two posterior probabilities of *S* (*=1*) and *R* (*=1*) in condition *W* (*=1*). These probabilities are also called *explanations* for *W*.

$$P(R=1 \mid W=1) = \frac{\sum_{C,S} P(C, R=1, S, W=1)}{\sum_{C,R,S} P(C, R, S, W=1)} = 0.581$$

$$P(S=1 \mid W=1) = \frac{\sum_{C,R} P(C, R, S=1, W=1)}{\sum_{C,R,S} P(C, R, S, W=1)} = 0.614$$

Because of $P(R=1|W=1) < P(S=1|W=1)$, it is concluded that sprinkler is the most likely cause of wet grass. Note that two above formulas which are also variants of Bayesian rule (see formula 2.1.1) will be surveyed more carefully in the "Bayesian network inference" section.

2.4 Markov Condition and Markov Equivalence

The inference in BN becomes complex and ineffective when the size of BN is large. Suppose BN has *n* binary nodes. In the worst case, each node has *n-1* parents, thus, the joint probability has $n*2^n$ entries. There is a boom of CPT (s). There is a restrictive criterion so-called Markov condition that makes the relationships (also CPT) among nodes simpler. Given Bayesian network (G, P) and three sets of nodes: $A=\{X_i, ..., X_j\}$, $B=\{X_k, ..., X_l\}$ and $C=\{X_m, ..., X_n\}$:
- The denotation $I_P(A,B)$ or $I_G(A,B)$ indicates that *A* and *B* are independent.
- The denotation $I_P(A,B|C)$ or $I_G(A,B|C)$ indicates that *A* and *B* conditional independent given *C*.

Let (*G*, *P*) be Bayesian network, Markov condition is stated that every node *X* is conditional independent from its non-descendants given its parent. In other word node *X* is only dependent on its previous nodes (directed parents).

$\forall X \in E$, $I_P(X, N_X \mid PA_X)$

where E is the set of edges in G, N_X and PA_X are set of non-descendants of X and parents of X, respectively.

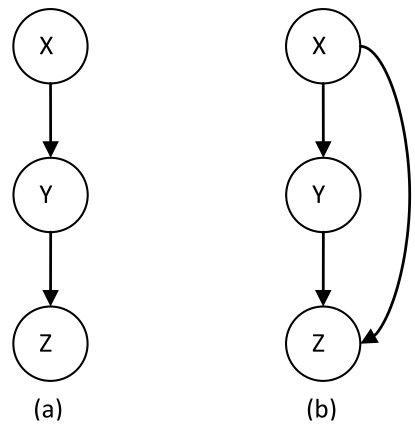


Figure 2.4.1: Example about Markov condition: (a) satisfy, (b) not satisfy

Because inference and structure learning algorithms are based on Markov condition, please pay attention to it.

Suppose Bayesian (G, P) satisfies Markov condition, it is necessary to find out or check whether a node (or a set of nodes) Z that separates a node (or a set of nodes) X from another node (or a set of nodes) Y. It means that whether there is $I_P(X, Y \mid Z)$. In this case, X and Y are called *d-separated* by Z.

There are some important concepts that constitute the d-separation concept:

- The chain $X \rightarrow Z \rightarrow Y$ or $X \leftarrow Z \leftarrow Y$ is called serial path.
- The chain $X \rightarrow Z \leftarrow Y$ is called convergent.
- The chain $X \leftarrow Z \rightarrow Y$ is called divergent.
- The chain X–Z–Y is called uncoupled chain if X and Y aren't adjacent.

Of course, serial path, convergent path and divergent path are uncoupled chain.

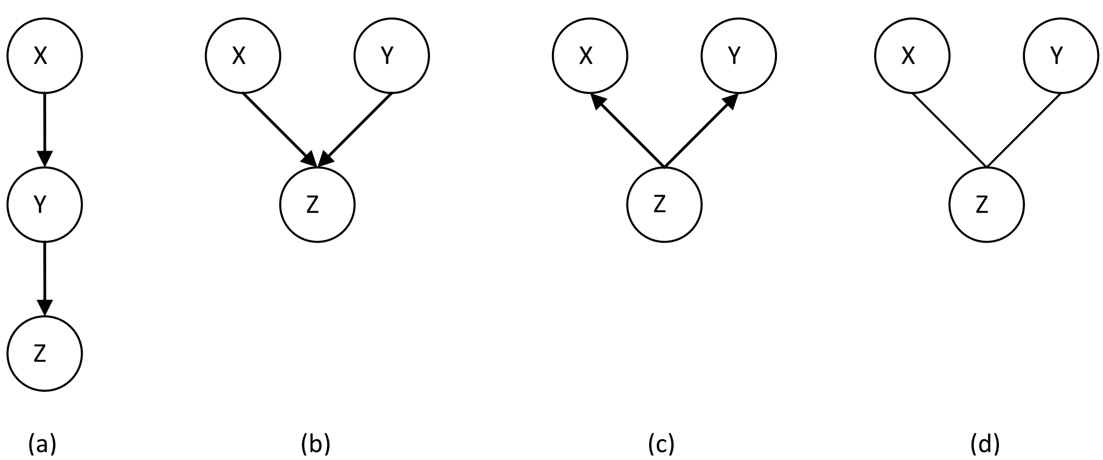
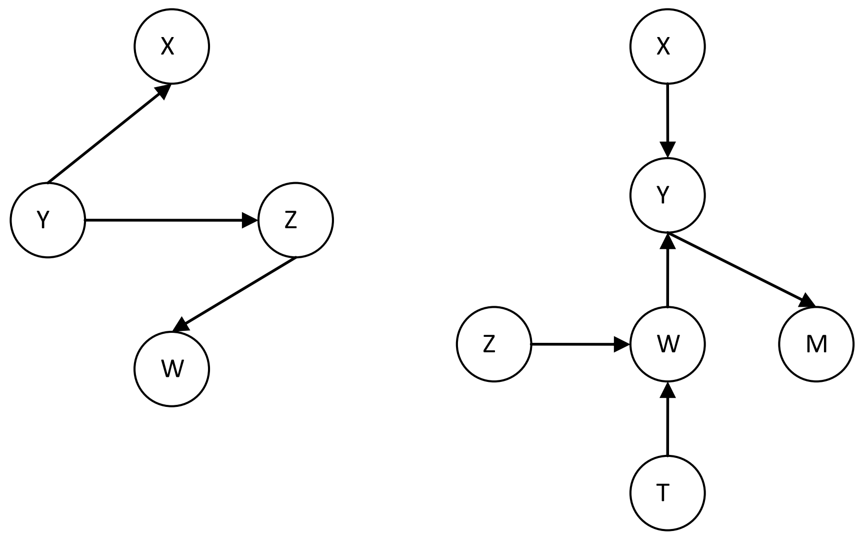


Figure 2.4.2: Serial path (a), convergent path (b), divergent path (c), and uncoupled chain (d)

Let X, Y and Z be sets of nodes where $X, Y, Z \subseteq V$. Given the chain p between X and Y, p is blocked by Z if and only if one of two conditions is satisfied:

International Open Journal of Operations Research
Vol. 1, No. 2, May 2013, PP: 16 – 39, ISSN: 2328-8582 (ONLINE)
Available online at http://www.acascipub.com/Journals.php

- There is an intermediate node $M \in Z$ on p so that all edges on p incident to M are serial or divergent at M.
- There is an intermediate node M on p so that:
  - $M \notin Z$ and all descendants of $M \notin Z$
  - All edges op p incident to M are convergent.



(a) (b)

Figure 2.4.3: The chain X–Y–Z–W in (a) is blocked by {Y, Z} because edges incident to Y are divergent at Y. The chain X–Y–Z–W–T in (b) is blocked by {Z, W} because there is such a node Y on chain that $Y \notin$ {Z, W}, its descendant $M \notin$ {Z, W}, and edges incident to Y are convergent at Y .

X and Y are d-separated by Z if all chains between X and Y are blocked by Z. Z is also called a *d*-separation of G.

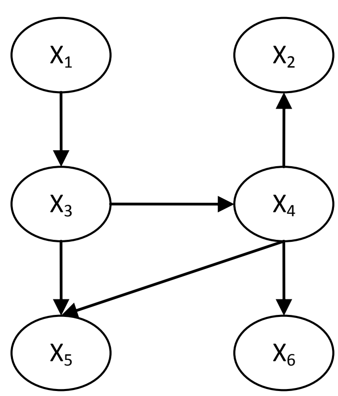


Figure 2.4.4: {X_1, X_2} is *d*-separated from {X_5, X_6} by {X_3, X_4}

BN (s) which have the same set of nodes are Markov equivalent if and only if they have same *d*-separations. In other words, BN (s) that are Markov equivalent have the same independences. Given G_1=(V, E_1) and G_2=(V, E_2), we have:

$$\forall A, B, C \subseteq V, I_{G_1}(A, B \mid C) \Rightarrow I_{G_2}(A, B \mid C)$$

where A, B, C are mutually disjoint sub-set of V. Note that G_1 and G_2 must be DAG and satisfy Markov condition.
The goal of giving "Markov equivalent" concept is to represent BN (s) that have the same structure and joint probability. So the representation of such BN (s) is called *Markov equivalent class* which is also a Bayesian network. In conclusion,

Markov equivalence divides all DAG (or BN) into disjoint *Markov equivalent classes*. In practice, Markov equivalent class is often find out or surveyed instead of considering many BN (s).

3. Bayesian Network Inference

3.1. Simple Inference

The essence of Bayesian reference is to compute the posterior probabilities of nodes given evidences. Note that evidences or conditions are also nodes which are observed and have concrete values. Back example 1.1 "wet grass". The posterior probability of $R = 1$ (rain) given $W = 1$ (wet grass) is the ratio of the marginal probability of *R*, *W* over *C*, *S* to the marginal probability of *W* over *C*, *R*, *S*.

$$P(R=1 \mid W=1) = \frac{P(R=1, W=1)}{P(W=1)} = \frac{\sum_{C,S} P(C, R=1, S, W=1)}{\sum_{C,R,S} P(C, R, S, W=1)}$$

Let $V=\{X_1, X_2, \ldots X_n\}$ be a whole set of nodes. Let $D=\{X_m, X_u, \ldots, X_n\}$ be a set of evidences, $D \subset V$. Let $d=(x_m, x_u, \ldots, x_n\}$ be the instantiation of *D*. In general case, the marginal probability of $X_k=x_k$ is:

$$P(X_k = x_k, D = d) = \sum_{V-\{X_k, D\}} P(X_1, X_2, \ldots, x_k, \ldots, d, \ldots, X_n)$$

where $P(X_1, X_2, \ldots, X_n)$ is the global joint probability.

The marginal probability of $D = d$ is:

$$P(D = d) = \sum_{V-D} P(X_1, X_2, \ldots, d, \ldots, X_n)$$

The probability of $X_k = k$ given $D = d$ is:

$$P(X_k = x_k \mid D = d) = \frac{P(X_k = x_k, D = d)}{P(D = d)} = \frac{\sum_{V-\{X_k, D\}} P(X_1, X_2, \ldots, x_k, \ldots, d, \ldots, X_n)}{\sum_{V-D} P(X_1, X_2, \ldots, d, \ldots, X_n)} \tag{3.1.1}$$

The above formula is the basic idea of simple inference. Note that it is also a variant of Bayesian rule (see formula 2.1.1). But the cost of computing it based on marginal probabilities is very high because there are a huge number of numeric operations such as additions and multiplications in computation expression. If the joint probability has many terms, brute force method for determining combinations of such operations is impossible. There are three main approaches that improve this computation:

- Taking advantage of Markov condition: Pearl's message propagation is well-known algorithm.
- OR-gate model inference which simulates OR-gate electronic circuit.
- Reducing the amount of numeric operations computed in marginal probability. Optimal factoring is the well-known technique.

3.2. Pearl's Message Propagation Algorithm

Suppose Bayesian network is DAG $G=(E, V)$ which is a tree having only one root. Given a set of evidence nodes $D \subseteq V$; every node in *D* has concrete value. Let D_X is the sub-set of *D* including *X* and descendants of *X* and let N_X be the sub-set of *D* including *X* and non-descendant of *X*. Let C_X and PA_X are children and parents of *X* respectively and *R* be root node and *O* be evidence node, $O \in D$.

International Open Journal of Operations Research
Vol. 1, No. 2, May 2013, PP: 16 – 39, ISSN: 2328-8582 (ONLINE)
Available online at http://www.acascipub.com/Journals.php

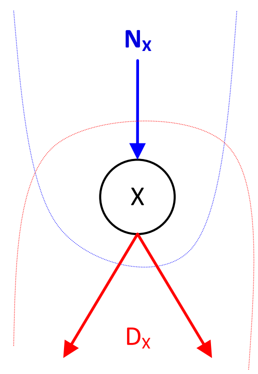


Figure 3.2.1: X, D_X and N_X. Note that N_X is green and D_X is red

The essence of inference is to compute the posterior probability $P(X|D)$ for every X. We have:

$$
\begin{aligned}
&P(X \mid D) = P(X \mid D_X, N_X) \\
&= \frac{P(D_X, N_X \mid X)P(X)}{P(D_X, N_X)} \text{ (due to Bayes' rule)} \\
&= \frac{P(D_X \mid X)P(N_X \mid X)P(X)}{P(D_X, N_X)} \text{ (} D_X \text{ and } N_X \text{ are conditionally independent given X)} \\
&= P(D_X \mid X)\frac{P(N_X \mid X)P(X)}{P(N_X)}\frac{P(N_X)}{P(D_X, N_X)} \\
&= \alpha P(D_X \mid X)P(X \mid N_X)
\end{aligned}
$$

where $\alpha = \frac{P(N_X)}{P(D_X, N_X)}$ is the constant independent from X.

Let $\lambda(X)$ and $\pi(X)$ be $P(D_X|X)$ and $P(X|N_X)$, respectively. Then we have

$$P(X|D) = \alpha\lambda(X)\ \pi(X) \tag{3.2.1}$$

The $\lambda(X)$ and $\pi(X)$ are called λ value and π value of X, respectively.

For each child Y of X, let $\lambda_Y(X)$ be λ message that connects X and Y. Note that $\lambda_Y(X)$ is conditional probability of D_Y given X.

$$\lambda_Y(X) = P(D_Y \mid X) = \sum_Y P(D_Y \mid Y)P(Y \mid X) = \sum_Y \lambda(Y)P(Y \mid X) \tag{3.2.2}$$

For each parent Z of X, let $\pi_X(Z)$ be π message that connects Z and X. Note that $\pi_X(Z)$ is conditional probability of X given N_X.

International Open Journal of Operations Research
Vol. 1, No. 2, May 2013, PP: 16 – 39, ISSN: 2328-8582 (ONLINE)
Available online at http://www.acascipub.com/Journals.php

$$\pi_X(Z) = P(Z \mid N_X)$$
$$= P(Z \mid N_Z, \bigcap_{K \in C_Z - \{X\}} D_K) \quad \text{(where } C_Z - \{X\} \text{ is the set of Z's children except X)}$$
$$= \frac{P(N_Z, \bigcap_{K \in C_Z - \{X\}} D_K \mid Z) P(Z)}{P(N_Z, \bigcap_{K \in C_Z - \{X\}} D_K)} \quad \text{(Bayes' rule)}$$
$$= \frac{P(N_Z \mid Z) P(\bigcap_{K \in C_Z - \{X\}} D_K \mid Z) P(Z)}{P(N_Z, \bigcap_{K \in C_Z - \{X\}} D_K)} \quad \text{(because Z and } C_Z - \{X\} \text{ are conditional independent given Z)}$$
$$= \frac{P(Z \mid N_Z) P(N_Z) P(\bigcap_{K \in C_Z - \{X\}} D_K \mid Z) P(Z)}{P(Z) P(N_Z, \bigcap_{K \in C_Z - \{X\}} D_K)}$$
$$= P(Z \mid N_Z) P(\bigcap_{K \in C_Z - \{X\}} D_K \mid Z) \frac{P(N_Z)}{P(N_Z, \bigcap_{K \in C_Z - \{X\}} D_K)}$$
$$= kP(Z \mid N_Z) P(\bigcap_{K \in C_Z - \{X\}} D_K \mid Z) \quad \text{(where } k = \frac{P(N_Z)}{P(N_Z, \bigcap_{K \in C_Z - \{X\}} D_K)} \text{ is the constant independent from X, Z)}$$
$$= k\pi(Z) \prod_{K \in C_Z - \{X\}} P(D_K \mid Z) \quad \text{(because X's children are mutually independent)}$$
$$= k\pi(Z) \prod_{K \in C_Z - \{X\}} \lambda_K(Z)$$
$$\approx \pi(Z) \prod_{K \in C_Z - \{X\}} \lambda_K(Z) \qquad (2.4)$$

Don't worry about $\pi_X(Z)$ is proportioned to $\pi(Z) \prod_{K \in C_Z - \{X\}} \lambda_K(Z)$ by removing constant k because the posterior probability $P(X|D)$ itself is also proportioned to $\lambda(X)$ and $\pi(X)$ via constant α. These constants will be eliminated when $P(X|D)$ is normalized. Now we have:

- Value $\lambda(X) = P(D_X|X)$
- Message $\lambda_Y(X) = P(D_Y \mid X) = \sum_Y \lambda(Y) P(Y \mid X)$ for each $Y \in C_X$
- Value $\pi(X) = P(X|N_X)$
- Message $\pi_X(Z) = P(Z \mid N_X) = \pi(Z) \prod_{K \in C_Z - \{X\}} \lambda_K(Z)$ for each $Z \in PA_X$.

The λ and π values are updated according to λ and π messages. Whenever evidence $O \in D$ occurs, Pearl's algorithm propagates downwards π message and propagates upwards λ message in order to update λ value and π value of each variable X so that the posterior probability $P(X|D)$ can be computed. The process of upwards-downwards propagation spreads over all variables of network.

International Open Journal of Operations Research
Vol. 1, No. 2, May 2013, PP: 16 – 39, ISSN: 2328-8582 (ONLINE)
Available online at http://www.acascipub.com/Journals.php

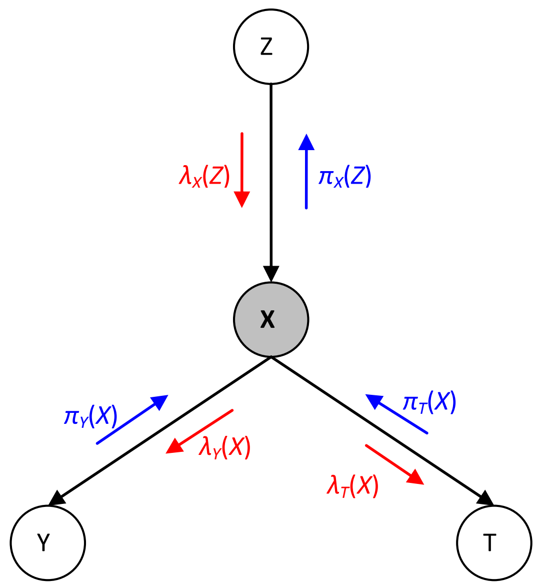


Figure 3.2.2: Pearl's propagation algorithm (*X* is focused node)

Please pay attention to following notices when updating λ value and π value at certain variable X:

1. If $X \in D$ and suppose X's instantiation (value) is x then:
   $\lambda(X=x) = P(x|x)= 1$ due to $X \in D_X$ and Markov condition. So $\lambda(X \neq x) = 0$
   $\pi(X=x) = P(x|x)= 1$ due to $X \in N_X$ and Markov condition. So $\pi(X \neq x) = 0$
   $P(X=x|D) = 1$ and $P(X \neq x|D) = 0$.
2. If $X \notin D$ and X is leaf then:
   $\lambda(X) = P(\varnothing|X) = 1$ due to $D_X = \varnothing$
   $\pi(X)$ is computed as if X were intermediate variable.
   $P(X|D)= \alpha\pi(X)$
3. If $X \notin D$ and X is root then:
   $\lambda(X)$ is computed as if X were intermediate variable.
   $\pi(X)=P(X|\varnothing)=P(X)$
   $P(X|D)= \alpha\lambda(X)P(X)$
4. If $X \notin D$ and X is intermediate variable then:
   $$\lambda(X) = P(D_X \mid X) = P(\bigcap_{Y \in C_X} D_Y \mid X) = \prod_{Y \in C_X} P(D_Y \mid X) = \prod_{Y \in C_X} \lambda_Y(X)$$
   (Because X's children are mutually independent)
   $$\pi(X) = P(X \mid N_X) = \sum_Z P(X \mid Z)P(Z \mid N_X) = \sum_Z P(X \mid Z)\pi_X(Z)$$
   where Z is parent of X.
   $P(X|D)= \alpha\lambda(X)\pi(X)$

Pseudo-code for Pearl's algorithm shown below includes four functions:

- Function "*void init*" initialize π value for every node. At that time the set of evidence nodes D is empty.
- Function "*void update*" is executed whenever evidence node O occurs. This function adds O to set D, propagates upwards λ message over all parents of O by calling function "*void propagate_up*", and propagates down π message over all children of O by calling function "*void propagate_down*".
- Function "*void propagate_up_λ_message*" computes λ value and posterior probability of current node, and continues to propagate upwards and downwards λ, π messages by calling itself and function "*void propagate_down_π_message*". Process of propagation stops when there is no node to be propagated.

- Function "*void propagate_down_π_message*" computes π value and posterior probability of current node, and continues to propagate downwards π message by calling itself. Process of propagation stops when there is no node to be propagated.

```
void init(G, D)
{
  D=Ø;
  for each X ∈ V
  {
    λ(X) = 1;                 //due to D = Ø
    for each parent Z of X    //propagate up λ message
      λX(Z) = 1;              // due to D = Ø
  }
  P(R|D) = P(R);              //posterior probability of root node
  π(R) = P(R);                // π value

  for each child K of R       //browse root's children
    propagate_up_π_message(R, K);
}

void update(O, o)
{
  D = D ∪ O
  λ(O=o) = π(O=o) = P(O=o|D) = 1;        //due to O ∈ D
  λ(O ≠ o) = π(O ≠ o) = P(O ≠ o|D) = 1;  //due to O ∉ D

  if O ≠ R and O's parent Z ∉ D          // O isn't root and parent of O doesn't belong to D
    propagate_up_ λ_message(O, Z);

  for each child K of O such that K ∉ D  //browse O's children
    propagate_up_π_message(O, K);
}

void propagate_up_λ_message(Y, X)
{
```

$$\lambda_Y(X) = \sum_Y \lambda(Y)P(Y \mid X)\text{;} \quad \text{//Y propagate upwards } \lambda \text{ message}$$

$$\lambda(X) = \prod_{Y \in C_X} \lambda_Y(X)\text{;} \quad \text{//update } \lambda \text{ value}$$

```
  P(X|D)= αλ(X)π(X);          //compute posterior probability of X
  normalize P(X|D);           //eliminate constant α

  if X ≠ R and X's parent Z ∉ D
    propagate_up_ λ_message(X, Z);

  for each child K of X such that K ≠ Y and K ∉ D  //browse O's children
    propagate_up_π_message(X, K);
}

void propagate_down_π_message(Z, X)
{
```

$$\pi_X(Z) = \pi(Z) \prod_{K \in C_Z - \{X\}} \lambda_K(Z)\text{;} \quad \text{//Y propagate downwards } \pi \text{ message}$$

International Open Journal of Operations Research
Vol. 1, No. 2, May 2013, PP: 16 – 39, ISSN: 2328-8582 (ONLINE)
Available online at http://www.acascipub.com/Journals.php

$\pi(X) = \sum_Z P(X \mid Z)\pi_X(Z)$; *//update* π *value*

P(X|D)= $\alpha\lambda(X)\pi(X)$*;* *//compute posterior probability of X*
normalize P(X|D); *//eliminate constant* α

for each child K of X such that $K \notin D$ *//browse O's children*
propagate_up_π_message(*X*, *K*)*;*
}

Example 3.2.1: Given Bayesian network shown in figure 3.2.3, suppose evidence *X* has value *1*. Hence, we need to compute posterior probabilities of *T, Y, Z* in condition *X=1*. Firstly, function "*void init*" is called to initialize network.

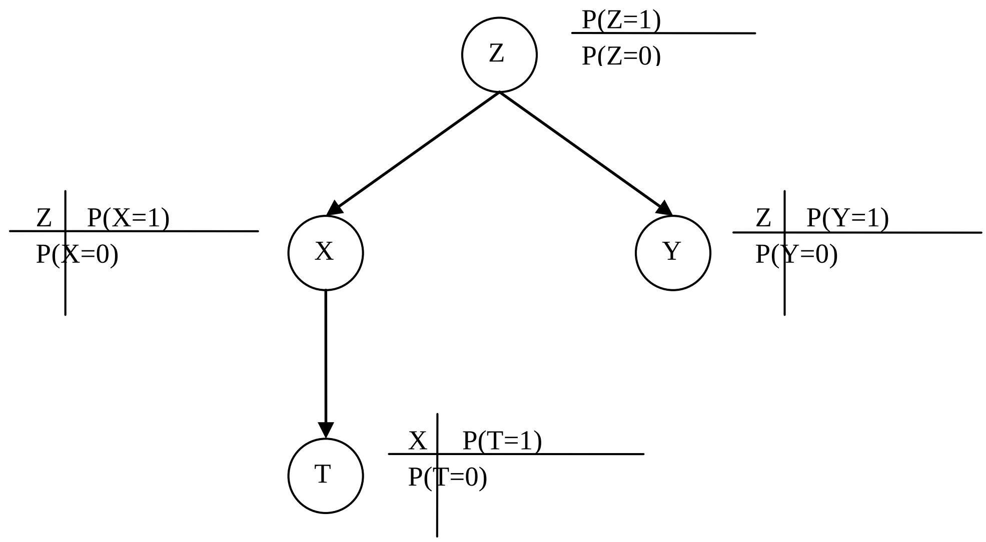


Figure 3.2.3: Bayesian network with CPT (s)

Function *init*(*G*,*D*) is executed:

$$D = \varnothing$$
$$\lambda(Z=1) = \lambda(Z=0) = 1$$
$$\lambda(X=1) = \lambda(X=0) = 1$$
$$\lambda(Y=1) = \lambda(Y=0) = 1$$
$$\lambda(T=1) = \lambda(T=0) = 1$$

$$\lambda_X(Z=1) = \lambda_X(Z=0) = 1$$
$$\lambda_Y(Z=1) = \lambda_Y(Z=0) = 1$$
$$\lambda_T(X=1) = \lambda_T(X=0) = 1$$

P(Z=1|d) = P(Z=1) = 0.6. Note that let d be instantiation of D
P(Z=0|d) = P(Z=0) = 0.4
$\pi(Z=1) = P(Z=1) = 0.6$
$\pi(Z=0) = P(Z=0) = 0.4$

Calling propagate_down_π_message(Z, X)
Calling propagate_down_π_message(Z, Y)

International Open Journal of Operations Research
Vol. 1, No. 2, May 2013, PP: 16 – 39, ISSN: 2328-8582 (ONLINE)
Available online at http://www.acascipub.com/Journals.php

Then, function propagate_down_π_message(Z, X) is executed:

$$\pi_X(Z=1)= \pi(Z=1)\ \lambda_X(Z=1)=1*0.6=0.6$$
$$\pi_X(Z=0)= \pi(Z=0)\ \lambda_X(Z=0)=1*0.4=0.4$$

$$\pi(X=1) = P(X=1|Z=1)\ \pi_X(Z=1) + P(X=1|Z=0)\ \pi_X(Z=0) = 0.7*0.6 + 0.2*0.4 = 0.5$$
$$\pi(X=0) = P(X=0|Z=1)\ \pi_X(Z=1) + P(X=0|Z=0)\ \pi_X(Z=0) = 0.3*0.6 + 0.8*0.4 = 0.5$$

$$P(X=1) = \alpha\ \lambda(X = 1)\ \pi(X=1)=\alpha 1*0.5=\alpha 0.5$$
$$P(X=0) = \alpha\ \lambda(X = 0)\ \pi(X=0)=\alpha 1*0.5= \alpha 0.5$$

$$P(X=1) = \frac{\alpha 0.5}{\alpha 0.5+\alpha 0.5} = 0.5$$
$$P(X=0) = \frac{\alpha 0.5}{\alpha 0.5+\alpha 0.5} = 0.5$$

Calling propagate_down_π_message(X, T)

Then, function propagate_down_π_message(X, T) is executed:

$$\pi_T(X=1)= \pi(X=1) =0.5$$
$$\pi_T(X=0)= \pi(X=0) =0.5$$

$$\pi(T=1) = P(T=1|X=1)\ \pi_T(X=1) + P(T=1|X=0)\pi_T(X=0) = 0.9*0.5 + 0.4*0.5 = 0.65$$
$$\pi(T=0) = P(T=0|X=1)\ \pi_T(X=1) + P(T=0|X=0)\pi_T(X=0) = 0.1*0.5 + 0.6*0.5= 0.4$$

$$P(T=1) = \alpha\ \lambda(T = 1)\pi(T=1) = \alpha 1*0.65= \alpha 0.65$$
$$P(T=0) = \alpha\ \lambda(T = 0)\pi(T=0) = \alpha 1*0.4= \alpha 0.4$$

$$P(T=1) = \frac{\alpha 0.65}{\alpha 0.65+\alpha 0.4} = 0.62$$
$$P(T=0) = \frac{\alpha 0.4}{\alpha 0.65+\alpha 0.4} = 0.38$$

Then function propagate_down_π_message(Z, Y) is executed:

$$\pi_Y(Z=1)= \pi(Z=1)\lambda_Y(Z=1) =1*0.6=0.6$$
$$\pi_Y(Z=0)= \pi(Z=0)\lambda_Y(Z=0)=1*0.4=0.4$$

$$\pi(Y=1) = P(Y=1|Z=1)\pi_X(Z=1) + P(Y=1|Z=0)\pi_X(Z=0) = 0.6*0.6 + 0.3*0.3 = 0.45$$
$$\pi(Y=0) = P(Y=0|Z=1)\pi_X(Z=1) + P(Y=0|Z=0)\pi_X(Z=0) = 0.3*0.4 + 0.8*0.7 = 0.68$$

$$P(Y=1) = \alpha\lambda(Y = 1)\pi(Y=1) = \alpha 1*0.45= \alpha 0.45$$
$$P(Y=0) = \alpha\lambda(Y = 0)\pi(Y=0) = \alpha 1*0.68= \alpha 0.68$$

$$P(Y=1) = \frac{\alpha 0.45}{\alpha 0.45+\alpha 0.68} = 0.4$$
$$P(Y=0) = \frac{\alpha 0.68}{\alpha 0.45+\alpha 0.68} = 0.6$$

The initialized Bayesian network is shown below:

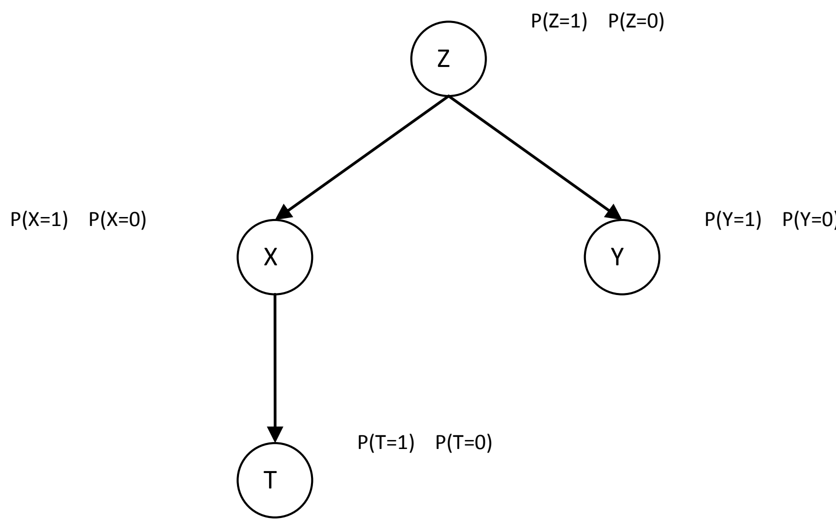


Figure 3.2.4: Initialized Bayesian network

When *X* becomes evidence and gains value *1*, the function *update*(*X*, *1*) is called:

$$D = D \cup X = \{X\}$$
Because d is instantiation of D, we have d = {X=1}
$$\lambda(X=1) = \pi(X=1) = P(X=1|d) = 1$$
$$\lambda(X=0) = \pi(X=0) = P(X=0|d) = 0$$

Calling propagate_up_λ_message(X, Z)
Calling propagate_down_π_message (X, T)

Then, function propagate_up_λ_message(X, Z) is executed:

$$\lambda_X(Z=1) = \lambda(X=1)P(X=1|Z=1) + \lambda(X=0)P(X=0|Z=1) = 1*0.7 + 0*0.3 = 0.7$$
$$\lambda(Z=1) = \lambda_X(Z=1)\lambda_Y(Z=1) = 0.7*1 = 0.7$$
$$P(Z=1|d) = \alpha\lambda(Z=1)\pi(Z=1) = \alpha 0.7*0.6 = \alpha 0.42$$

$$\lambda_X(Z=0) = \lambda(X=1)P(X=1|Z=0) + \lambda(X=0)P(X=0|Z=0) = 1*0.2 + 0*0.8 = 0.2$$
$$\lambda(Z=0) = \lambda_X(Z=0)\lambda_Y(Z=0) = 0.2*1 = 0.2$$
$$P(Z=0|d) = \alpha\lambda(Z=0)\ \pi(Z=0) = \alpha 0.2*0.4 = \alpha 0.08$$

$$P(Z=1|d) = \frac{\alpha 0.42}{\alpha 0.42 + \alpha 0.08} = 0.84$$
$$P(Z=0|d) = \frac{\alpha 0.08}{\alpha 0.42 + \alpha 0.08} = 0.16$$
Calling propagate_down_π_message(Z, Y)

Then, function propagate_down_π_message (Z, Y) is executed:

$$\pi_Y(Z=1) = \pi(Z=1)\ \lambda_Y(Z=1) = 1*0.6 = 0.6$$
$$\pi_Y(Z=0) = \pi(Z=0)\ \lambda_Y(Z=0) = 1*0.4 = 0.4$$

$$\pi(Y=1) = P(Y=1|Z=1)\ \pi_X(Z=1) + P(Y=1|Z=0)\ \pi_X(Z=0) = 0.6*0.6 + 0.3*0.4 = 0.48$$
$$\pi(Y=0) = P(Y=0|Z=1)\ \pi_X(Z=1) + P(Y=0|Z=0)\ \pi_X(Z=0) = 0.3*0.6 + 0.8*0.4 = 0.5$$

$$P(Y=1) = \alpha\ \lambda(Y = 1)\ \pi(Y=1) = \alpha 1*0.48= \alpha 0.48$$
$$P(Y=0) = \alpha\ \lambda(Y = 0)\ \pi(Y=0) = \alpha 1*0.5= \alpha 0.5$$

$$P(Y=1) = \frac{\alpha 0.48}{\alpha 0.48 + \alpha 0.5} = 0.49$$
$$P(Y=0) = \frac{\alpha 0.5}{\alpha 0.48 + \alpha 0.5} = 0.51$$

Then function propagate_down_π_message(X, T) is executed

$$\pi_T(X=1)= \pi(X=1) =1$$
$$\pi_T(X=0)= \pi(X=0) =0$$

$$\pi(T=1) = P(T=1|X=1)\ \pi_T(X=1) + P(T=1|X=0)\ \pi_T(X=0) = 0.9*1 + 0.4*0 = 0.9$$
$$\pi(T=0) = P(T=0|X=1)\ \pi_T(X=1) + P(T=0|X=0)\ \pi_T(X=0) = 0.1*1 + 0.6*0= 0.1$$

$$P(T=1) = \alpha\ \lambda(T = 1)\ \pi(T=1) = \alpha 1*0.9= \alpha 0.9$$
$$P(T=0) = \alpha\ \lambda(T = 0)\ \pi(T=0) = \alpha 1*0.1= \alpha 0.1$$

$$P(T=1) = \frac{\alpha 0.9}{\alpha 0.9 + \alpha 0.1} = 0.9$$
$$P(T=0) = \frac{\alpha 0.1}{\alpha 0.9 + \alpha 0.1} = 0.1$$

Finally, all posterior probabilities are computed as in following figure

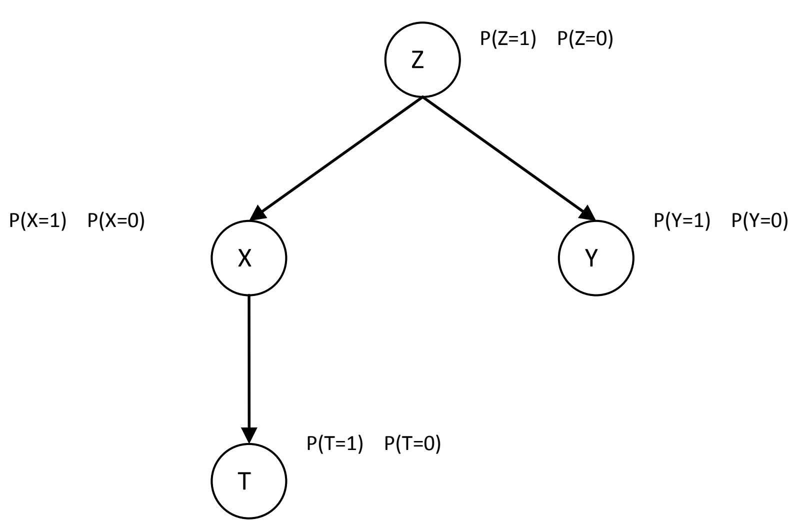


Figure 3.2.5: All posterior probabilities are computed after running Pearl algorithm (*X* is evidence)

3.3. OR-Gate Inference

In OR-gate electric circuit, the output value becomes *TRUE* if there is at least one of inputs being *TRUE*. Suppose every node is binary, OR-gate inference in Bayesian network simulates such circuit based on three assumptions:

- *Cause inhibition*: Given a cause-effect relationship denoted by edge $X \rightarrow Y$, there is a factor I that inhibits X from causing Y. Factor I is called inhibition of X. That the inhibition I is turned off is the prerequisite of X causing Y.
  $I = 0 \Leftrightarrow I \text{ turned OFF}$

  $I = 1 \Leftrightarrow I \text{ turned ON}$
- *Inhibition independence*: Inhibitions are mutually independent. For example inhibition I_1 of X_1 is independent from inhibition I_2 of X_2.
- *OR condition*: Suppose we have a set of cause-effect relationships in which Y is the effect of many causes X_1, $X_2, \ldots, X_n$ (see following figure). Let I_i be the inhibition of X_i. The effect Y *cannot happen* ($Y=0$) if *at least* one of X_i is equal *0* or one of inhibitions is *ON*:
  $\exists i : X_i = 0 \vee I_i = 1 \Rightarrow Y = 0$

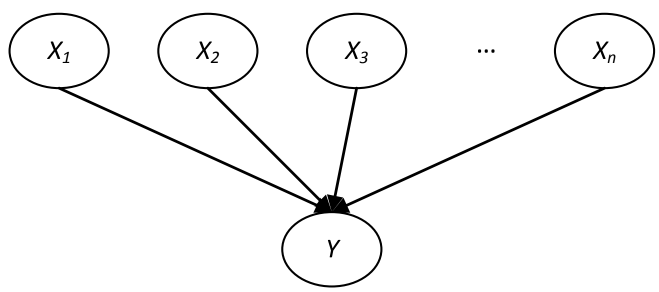


Figure 3.3.1: Cause-effect relationships

Suppose we have n causes $X_1, X_2, \ldots, X_n$ and one result Y. According to "*cause inhibition*" and "*inhibition independence*" assumptions, let I_i be the inhibition of X_i. Let A_i be dummy variable so that A_i is *ON* (=*1*) if X_i is equal to *1* and I_i is *OFF* (=0).

$$P(A_i = ON \mid X_i=1, I_i=OFF) = 1$$
$$P(A_i = ON \mid X_i=1, I_i=ON) = 0$$
$$P(A_i = ON \mid X_i=0, I_i=OFF) = 0$$
$$P(A_i = ON \mid X_i=0, I_i=ON) = 0$$

$$P(A_i = OFF \mid X_i=1, I_i=OFF) = 0$$
$$P(A_i = OFF \mid X_i=1, I_i=ON) = 1$$
$$P(A_i = OFF \mid X_i=0, I_i=OFF) = 1$$
$$P(A_i = OFF \mid X_i=0, I_i=ON) = 1$$

Applying "*OR condition*", the condition probability of Y is equal *0* (Y never happens) if at least one A_i is *ON*. It means that Y happens ($Y=1$) if all A_i (s) are *ON*.

$$P(Y=0 \mid \exists A_i=ON) = 0$$
$$P(Y=0 \mid \forall A_i=OFF) = 1$$
$$P(Y=1 \mid \forall A_i=ON) = 1$$
$$P(Y=1 \mid \exists A_i=OFF) = 0$$

International Open Journal of Operations Research
Vol. 1, No. 2, May 2013, PP: 16 – 39, ISSN: 2328-8582 (ONLINE)
Available online at http://www.acascipub.com/Journals.php

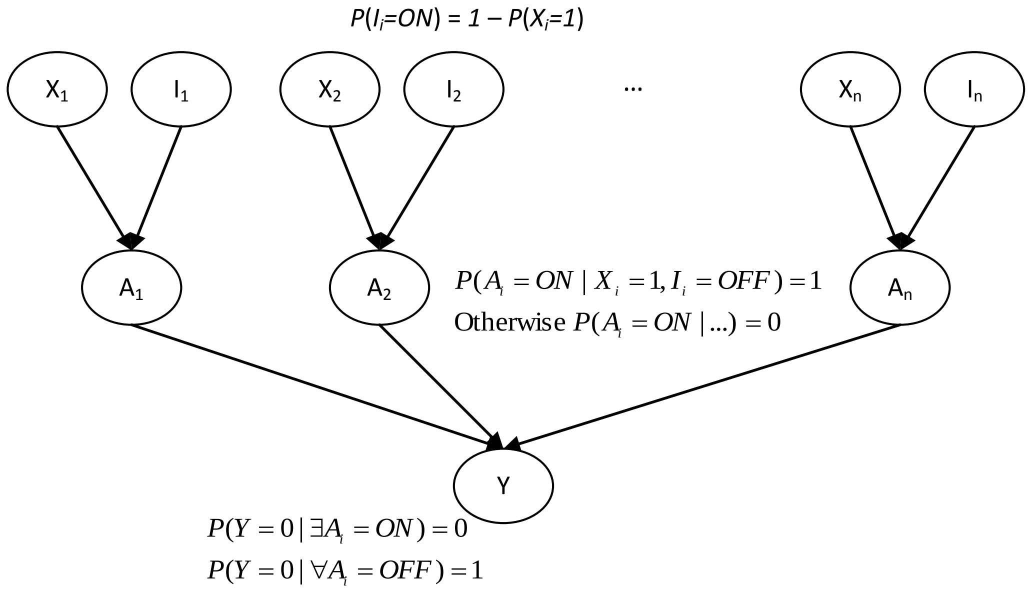


Figure 3.3.2: OR-gate model

Now the strength of each cause-effect relationship $X_i \rightarrow Y$ is quantified by the CPT $P(Y|X_i)$. Suppose causes $(X_1, X_2, \ldots, X_i, \ldots, X_n)$ become evidences having values $(x_1, x_2, \ldots, x_i, \ldots, x_n)$. Let $P(X_i=1) = p_i$ be the probability of $X_i = 1$. The probability of X_i 's inhibition is the inverse:

$$P(I_i=ON) = 1 - P(X_i=1)$$

Let O be the set of such i that $X_i = 1$.

$$\forall i \in O, X_i = 1$$

The goal of inference is to determine the posterior probability $P(Y| X_1, X_2, \ldots, X_i, \ldots, X_n)$. We have:

$$\begin{aligned}
&P(Y = 0 \mid X_1 = x_1, \ldots, X_n = x_n) \\
&= \sum_{a_1, \ldots, a_n} P(Y = 0 \mid A_1 = a_1, \ldots, A_n = a_n) P(A_1 = a_1, \ldots, A_n = a_n \mid X_1 = x_1, \ldots, X_n = x_n) \\
&\text{(Due to the law of total probability)} \\
&= \sum_{a_1, \ldots, a_n} P(Y = 0 \mid A_1 = a_1, \ldots, A_n = a_n) \prod_i P(A_i \mid X_1 = x_1, \ldots, X_n = x_n) \\
&\text{(Due to } \mathrm{A_i} \text{ (s) are mutually independent)} \\
&= \sum_{a_1, \ldots, a_n} P(Y = 0 \mid A_1 = a_1, \ldots, A_n = a_n) \prod_i P(A_i \mid X_i = x_i) \\
&\text{(Because } \mathrm{A_i} \text{ is only dependent on } \mathrm{X_i}\text{)} \\
&= \prod_i P(A_i = OFF \mid X_i = x_i) \\
&\text{(Because that any } \mathrm{A_i} \text{ is equal ON causes the conditional probability } \mathrm{P(Y = 0 \mid A_i = ON) = 0}, \\
&\text{we just focus on } \mathrm{A_i = OFF}\text{)}
\end{aligned}$$

$$= \prod_{i} (P(A_i = OFF \mid X_i = x_i, I_i = ON)P(I_i = ON) + P(A_i = OFF \mid X_i = x_i, I_i = OFF)P(I_i = OFF))$$
$$= \prod_{i \in O} (P(A_i = OFF \mid X_i = 1, I_i = ON)P(I_i = ON) + P(A_i = OFF \mid X_i = 1, I_i = OFF)P(I_i = OFF))$$
$$+ \prod_{i \notin O} (P(A_i = OFF \mid X_i = 1, I_i = ON)P(I_i = ON) + P(A_i = OFF \mid X_i = 1, I_i = OFF)P(I_i = OFF))$$
$$= \prod_{i \in O} (1(1 - P(X_i)) + 0P(X_i)) \prod_{i \notin O} (1(1 - P(X_i)) + 1P(X_i))$$
$$= \prod_{i \in O} (1 - P(X_i))$$

In conclusion, we have

$$P(Y=0|X_1,X_2,\ldots, X_n) = \prod_{i \in O} (1 - P(X_i)) \quad (3.3.1)$$

$$P(Y=1|X_1,X_2,\ldots, X_n) = 1 - \prod_{i \in O} (1 - P(X_i)) \quad (3.3.2)$$

Where O is the set of such i that $X_i = 1$.

Example 3.3.1: Given cause-effect relationship shown in following figure. Given prior probabilities of causes X_1, X_2, X_3 are 0.2, 0.5, 0.3, respectively. We need to compute the conditional probability of effect $P(Y=1|X_1=1, X_2=0, X_3=1)$.

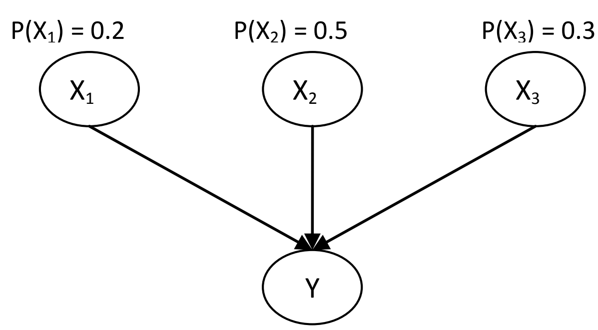


Figure 3.3.3: Example of OR-gate inference

Applying formula 3.3.1, we have:

$$P(Y=1| X_1=1, X_2=0, X_3=1) = 1 - (1 - P(X_1=1))(1 - P(X_3=1)) = 1 - 0.8*0.7 = 0.44$$

4. Optimal Factoring Technique

The basic idea of optimal factoring technique is to reduce the amount of numeric operations by changing the order of combinations of such operations. Back example 2.2.1, given joint probability $P(C, R, S, W)=P(C)*P(S)*P(R|C)*P(W|R,S)$, the marginal probability of $R = 1$ is factorized as below:

$$P(R=1, W=1) = \sum_{C,S} P(C)P(S)P(R=1|C)P(W=1|R=1,S)$$

International Open Journal of Operations Research
Vol. 1, No. 2, May 2013, PP: 16 – 39, ISSN: 2328-8582 (ONLINE)
Available online at http://www.acascipub.com/Journals.php

Because each binary variable has *2* values, there are 2^2 combinations of *C* and *S*. Each product has *3* multiplications. So the total number of required multiplications is $3*2^2 = 12$. Now the ordering of expression is changed by the factorization as below:

$$P(R=1,W=1)=\sum_C P(C)P(R=1|C)\sum_S P(S)P(W=1|R=1,S)$$

The inner sum of products $\sum_S P(S)P(W=1|R=1,S)$ has $1*2^1=2$ multiplications. Although the outer sum of products $\sum_C P(C)P(R=1|C)\sum_S(...)$ contains *4* variables, it has $2*2^1=4$ multiplications because expressions which don't relate to variable *S* such as *P*(*C*) and *P*(*R=1*|*C*) are taken out the inner sum of products. So the total number of required multiplications is *4+2=6*. Six multiplications are saved.

It is easy to recognize the best ordering of expressions which produces the minimal required multiplications if the number of variables is small. How we can do that in case of many variables. The answer relates to the optimal factoring problem.

Given *F* = (*V*, *S*, *Q*) is defined as the triple consisting of Richard [7, pp. 163]:

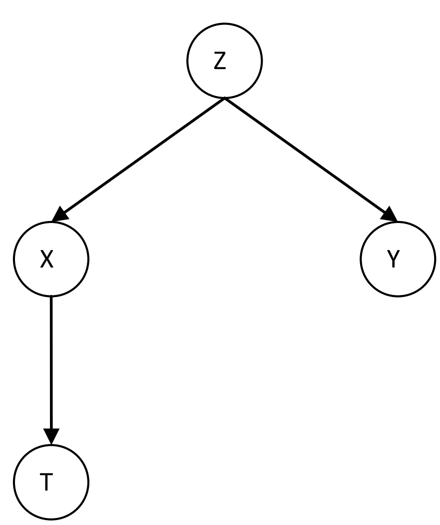
- A set of *n* nodes (or variables) $V=\{X_1, X_2, ..., X_n\}$
- A set of *m* sub-sets $S = \{S_{\{1\}}, S_{\{2\}}, ..., S_{\{m\}}\}$ where $S_{\{I\}} \subset V$
- A *target set* $Q \subset V$

The factoring α of *S* is a binary tree satisfying three following condition as in Richard [7, pp.164]:

- All and only member $S_{\{I\}}$ of S are leaves.
- The parent of nodes $S_{\{I\}}$ and $S_{\{J\}}$ are denoted $S_{\{I \cup J\}}$
- The root of tree is $S_{\{1,2,..,m\}}$

Note that *S* corresponds to operands of marginal probability and α corresponds with the factorization of marginal probability.

Example 4.1: Like example 3.3.1, let *Z, X, Y, T* be nodes of Bayesian network shown in following figure 4.1.



International Open Journal of Operations Research
Vol. 1, No. 2, May 2013, PP: 16 – 39, ISSN: 2328-8582 (ONLINE)
Available online at http://www.acascipub.com/Journals.php

The joint probability is $P(Z,X,Y,T) = P(Z)P(X|Z)P(Y|Z)P(T|X)$. Suppose X is evidence, we need to compute the posterior conditional probability $P(Z=1|X=1)$. The marginal probability over Z, X shown below is the sum of products which will be optimized:

$$P(Z=1, X=1) = \sum_{Y,T} P(Z=1)P(X=1|Z=1)P(Y|Z=1)P(T|X=1)$$

The factoring instance $F(V, S, Q)$ is defined as below:

- $V = \{Z, X, Y, T\}$
- $S = \{S_{\{1\}}=\{Z\}, S_{\{2\}}=\{X, Z\}, S_{\{3\}}=\{Y, Z\}, S_{\{4\}}=\{T,X\}\}$
- $Q = \{Z, X\}$

Suppose factoring α_1, α_2 correspond to two factorizations of marginal probability $P(Z=1,X=1)$.

$$\alpha_1 \approx P(Z=1)P(X=1|Z=1)\sum_{Y}(P(Y|Z=1)\sum_{T}P(T|X=1)$$

$$\alpha_2 \approx \sum_{Y,T} P(Z=1)P(X=1|Z=1)P(Y|Z=1)P(T|X=1)$$

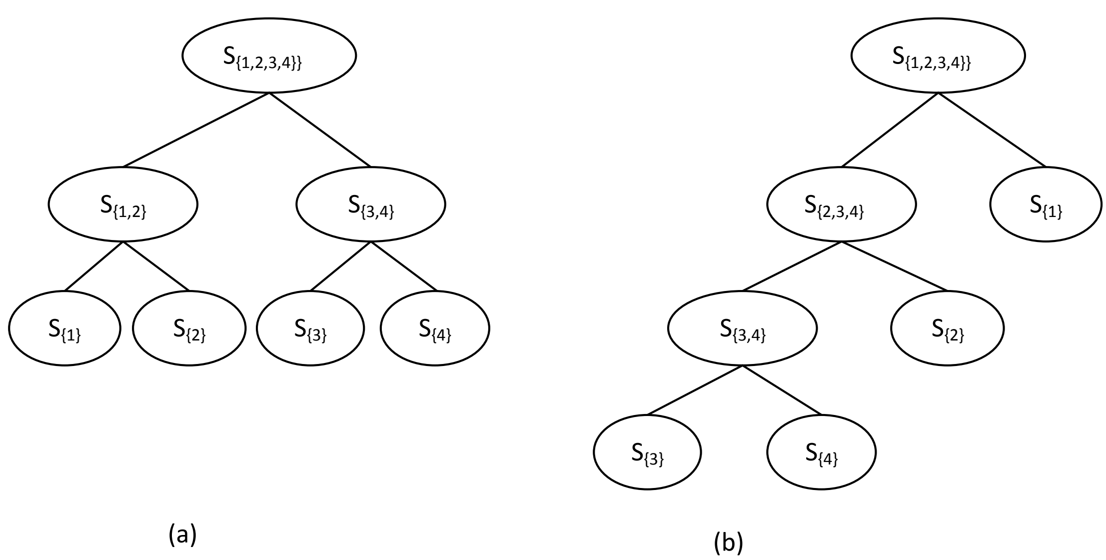


Figure 4.2: (a) Factoring α_1 and (b) Factoring α_2

Given F, the cost of factoring α denoted $cost_\alpha(F)$ is two following steps:

1. **Step 1.** All non-leave nodes are determined according to formula:
   $S_{\{I \cup J\}} = S_{\{I\}} \cup S_{\{J\}} - W_{\{I \cup J\}}$ where $W_{\{I \cup J\}} = \{w \notin Q$ and $w \notin S_{\{k\}}$ for all $k \notin I \cup J\}$

2. **Step 2.** The cost of each node is computed according to formula:

For leaf nodes: $cost_\alpha(S_{\{j\}}) = 0, j=\overline{1,m}$

For non-leaf nodes: $cost_\alpha(S_{\{I \cup J\}}) = cost_\alpha(S_I) + cost_\alpha(S_J) + 2^{|S_I \cup S_J|}$

where |.| denotes the cardinality of the set.

The cost of factoring α: $cost_\alpha(F)= cost_\alpha(S_{\{1,\ldots,m\}})$. The less this cost is, the better binary tree is.

Applying optimal factoring problem into Bayesian inference, the set of nodes V in F corresponds with variables in BN and the tree α corresponds with the ordering of multiplications in marginal probability. The cost of factoring instance $cost_\alpha(F)$ is equal to the number of multiplications. The problem becomes easy when we find out the best binary tree α having the least $cost_\alpha(F)$ and compute the marginal probability with the same ordering of multiplications to this tree.

Back example 3.3.1, the cost of factoring α_1 is computed as below:

$$cost_{\alpha 1}(S\{1,2,3,4\}) = cost_{\alpha 1}(S_{\{1,2\}}) + cost_{\alpha 1}(S_{\{3,4\}}) = (0+0+2^0) + (0+0+2^2) = 5$$

$$cost_{\alpha 2}(S\{1,2,3,4\}) = cost_{\alpha 1}(S_{\{2,3,4\}}) + cost_{\alpha 1}(S_{\{1\}}) + 2^2 = cost_{\alpha 1}(S_{\{2,3,4\}}) + 0 + 2^2$$

$$= cost_{\alpha 1}(S_{\{3,4\}}) + cost_{\alpha 1}(S_{\{2\}}) + 2^2 + 2^2$$

$$= (0+0+2^1) + 0 + 2^2 + 2^2 = 10$$

Because $cost_{\alpha 1}(S\{1,2,3,4\})$ is lesser than $cost_{\alpha 2}(S\{1,2,3,4\})$, the following ordering of multiplications is chosen:

$$P(Z=1, X=1) = P(Z=1)P(X=1 \mid Z=1)\sum_Y (P(Y \mid Z=1)\sum_T P(T \mid X=1)$$

5. Conclusions and Policy Recommendations

In this paper, inference mechanism as a significant domain of Bayesian network (BN) is explored exhaustively. Different components of inference mechanism are described with numerical illustrations. Since the inference mechanism pays a vital role to communicate the usability of Bayesian network, therefore, keeping in view the gravity of the inference mechanism we have paid special attention to explore it exhaustively. Briefly we draw following some more conclusions after a little review of the present paper:

- The ideology of Bayesian network is to apply a mathematical inference tool (namely Bayesian rule) into a graph with expectation of extending and enhancing the ability of such tool so as to sole realistic problems, especially diagnosis domain. Pearl's message propagation algorithm in connection to Bayesian network inference has also been depicted in section 3.2. The posterior probabilities are computed after running Pearl's message propagation algorithm.
- In view of recent work done by Maurya [3] in the process of developing Bayesian network; it has been observed that there arise many problems in continuous case and nodes dependency. In this article, we have focused on discrete case only when the probability of each node is discrete CPT, not continuous PDF.
- The optimal factoring technique has also been applied in section 4 in order to reduce the amount of numeric operations by changing the order of combinations of operations.
- It has been examined that the best ordering of expressions produces the minimal required multiplications if the number of variables is small. In case of many variables, the optimal factoring technique is considerably useful.
- Finally, it is remarked that the inference mechanism in Bayesian network is the key domain. Without inference mechanism, other two significant domains of Bayesian network namely parameter and structure learning are hardly possible.

Acknowledgements

Dr. V. N. Maurya; principal author of the present paper and former founder Director at Vision Institute of Technology, Aligarh (Uttar Pradesh Technical University, Lucknow (India)**,** former Principal/Director at Shekhawati Engineering College (Rajasthan Technical University, Kota) and former Professor & Dean Academics, Institute of Engineering & Technology, Sitapur, UP, India; is now the Professor & Head of Shekhawati Engineering College (Rajasthan Technical University, Kota). He is the Chief Editor of Editorial Board of American Journal of Modeling and Optimization; Science and Education Publishing, New York, USA and Statistics, Optimization and Information Computing; International Academic Press, Hong Kong and Advisory Editor of World Research Journal of Numerical Analysis and Mathematical Modeling; Bioinfo Publications, Pune, India and Member of Editorial and Reviewer Board of over 50 Indian and Foreign International journals published by leading publishers of USA, Italy, Hong Kong, Austria, U.K., Algeria, Nigeria and other European and African countries. He has been associated with leading Indian Universities-U. P. Technical University, Lucknow during 2005-06 and Chhatrapati Shahu Ji Maharaj University, Kanpur for three terms during 2000-2004 for significant contribution of his supervision as Head Examiner of Central Evaluation for Theory Examinations of UG (B.Tech./B.Pharm.) and PG (MA/M.Sc.) programmes.

During his tenure as the Director, Vision Institute of Technology, Aligarh (Uttar Pradesh Technical University, Lucknow) and as the Principal, Shekhawati Engineering College (Rajasthan Technical University, Kota); massive expansion of infrastructure, research facilities, laboratories upgradation/augmentation and other relevant facilities and services for B.Tech./M.Tech./MBA academic programmes in different branches had taken place to accommodate and facilitate the campus students. His major contribution was to enhance the result of weaker students of their University Examination. He planned strategically and developed some tools and methods and then finally implemented for getting successfully considerable better result of campus students particularly in numerical papers.

Dr. Maurya was born on 15th July 1974 in District Basti of U.P., India and he is having an outstanding academic record. He earned his M.Sc. and Ph.D. Degree in Mathematics & Statistics with specialization in Operations Research with First Division from Dr. Ram Manohar Lohia Avadh University, Faizabad, UP, India in the year 1996 and 2000 respectively and thereafter he accomplished another two years Master's Professional Degree-MBA with First Division (B+ Grade) with specialization in Computer Science from NU, California, USA in 2003. His Ph.D. Thesis titled as "A study of use of stochastic processes in some queueing models" submitted to Department of Mathematics & Statistics, Dr. R.M.L. Avadh University, Faizabad under supervision of Prof. (Dr.) S.N. Singh, Ph.D. (BHU); was offered to publish in Scholar's Press Publishing Co., Saarbrucken, Germany in view of his excellent research work. Since his primary education to higher education, he has been a meritorious scholar and recipient of meritorious scholarship. He started his teaching career as Lecturer in 1996 to teach post-graduate courses MBA, MCA and M.Sc. and later he was appointed as Professor & Head, Department of Applied Sciences and Engineering at Singhania University, Rajasthan in the year 2004. Since then, Prof. V. N. Maurya has rendered his services as Professor & Head/Dean as well as keen Researcher for Post-Doctoral research and he has devoted his entire scientific and professional career in teaching at various premier technical institutions of the country such as at Haryana College of Technology & Management, Kaithal (Kuruchhetra University, Kuruchhetra); Institute of Engineering & Technology, Sitapur and United College of Engineering & Research, Allahabad. On the basis of significant research work carried out by him in the last 17 years of his professional career, Prof. V. N. Maurya has authored three textbooks and published more than 55 scientific and academic research papers including 25 research papers as Principal Author based on his Post-Doctoral work and D.Sc. Thesis in Indian and Foreign leading International Journals in the field of Mathematical and Management Sciences, Industrial Engineering & Technology. Some of his published research papers in India, USA, Algeria, Malaysia and other European and African countries are recognized as innovative contributions in the field of Mathematical and Physical Sciences, Engineering & Technology. Prof. V. N. Maurya is known as one of the Indian leading experts in Mathematical & Physical Sciences and Operational Research on his significant contributions to many mathematical, statistical, computer science and industrial engineering related areas basic as well as application oriented. He is an approved Supervisor of UGC recognized various Indian Universities for Research Programs leading to M. Phil. & Ph.D. such as Shridhar University, Pilani (Rajasthan), Singhania University, Rajasthan and CMJ University, Sillong, Meghalaya and JJT University Jhunjhunu, Rajasthan and U.P. Technical University Lucknow etc. and since last 7 years, he is actively engaged as Research Supervisor of M. Phil. & Ph.D. Scholars in wide fields of Operations Research, Optimization Techniques, Statistical Inference, Applied

Mathematics, Operations Management and Computer Science. He has guided as Principal Supervisor and Co-Supervisor to several Research Scholars of M. Phil. and Ph.D.
Prof. Maurya is also on active role of Fellow/Senior/Life Member of various reputed National and International professional bodies of India and abroad including Operations Research Society of India, Kolkata; Indian Society for Technical Education, New Delhi; Indian Association for Productivity, Quality & Reliability, Kolkata; Indian Society for Congress Association, Kolkata; International Indian Statistical Association, Kolkata; All India Management Association, New Delhi; Rajasthan Ganita Parishad, Ajmer and International Association of Computer Science & Information Technology, Singapore etc.

Diwinder Kaur Arora; co-author of the present paper accomplished MBA Degree with specialization in Human Resources from Pondicherry Central University, Pondicherry and she graduated with B.Sc. (Medical/ZBC Group) Degree in 1987 from Kanpur University, Kanpur, India and did Diploma also from Government Polytechnic College, Amethi, U.P. throughout in First Division. She has vast experience of more than 22 years of general administration and management as Police Officer of Central Reserve Police Force, Ministry of Home Affairs, Govt. of India. She was selected as Assistant Sub-Inspector (Non-Gazetted Officer) in 1991 and after successful completion of her services she was promoted as Sub-Inspector in 2004 and since 2012 she is working in the grade of Inspector of Police at Group Centre, Central Reserve Police Force, Lucknow, U.P. Apart from this, she has published more than 12 research papers in Indian and Foreign International journals of repute in the field of Management, Information Technology and Physical Sciences such as in World of Sciences Journal, Engineers Press Publishing Group, Vienna, Austria; International Journal of Engineering Research and Technology, Engineering Science & Research Support Academy (ESRSA), Vadodara, India; International Journal of Electronics Communication and Electrical Engineering, Algeria; International Journal of Information Technology & Operations Management, Academic and Scientific Publisher, New York, USA.

Er. Avadhesh Kumar Maurya; co-author of the paper is having an outstanding academic record and accomplished his M.Tech. Degree with specialization in Digital Communication from Uttarakhand Technical University, Dehradun, UK and he was graduated with B.Tech. Degree in Electronics and Communication Engineering from Rajasthan Technical University, Kota (Rajasthan). He is recipient of four First Divisions in his Student Career with flying colours. Since last one year, Er. A. K. Maurya is serving as Assistant Professor in Department of Electronics and Communication Engineering at Lucknow Institute of Technology, U.P. Technical University, Lucknow. Prior to assuming the post of Assistant Professor at Lucknow Institute of Technology, U.P., he served as a Network Engineer for two years in National Informatics Centre, Department of Information Technology, Govt. of India with collaboration of HCL Co. He has worked on some projects such as Movable Target Shooter using Ultrasonic Radar and Hartley Oscillator. Apart from this, he has got industrial training in Door Darshan Kendra, Lucknow, U.P. in the field of TV Program Generation and Broadcasting of different channels for partial fulfilment of his Degree and published also over 15 research papers in various Indian and Foreign International journals of repute in the field of Electronics & Communication Engineering, Computer Science & Information Technology and Physical Sciences such as in International Journal of Electronics Communication and Electrical Engineering, Algeria; World of Sciences Journal, Engineers Press Publishing Group, Vienna, Austria; International Journal of Information Technology & Operations Management, Academic and Scientific Publisher, New York, USA; International Journal of Engineering Research and Technology, Engineering Science & Research Support Academy (ESRSA), Vadodara, India; International Journal of Software Engineering & Computing, Serials Publications, New Delhi, India and many more.

References

[1] David Heckerman, A tutorial on learning with Bayesian networks, Technical Report MSR-TR-95-06, Microsoft Research Advanced Technology Division, Microsoft Corporation

[2] Maurya V. N., Arora Diwinder Kaur, Maurya A. K. & Gautam R.A., Exact modelling of annual maximum rainfall with Gumbel and Frechet distributions using parameter estimation techniques, World of Sciences Journal, Engineers Press Publishing Group, Vienna, Austria, Vol. 1, No. 2, 2013, pp.11-26

[3] Maurya V. N., Arora Diwinder Kaur & Maurya A.K., A survey report of parameter and structure learning in Bayesian network inference, International Journal of Information Technology & Operations Management, Academic & Scientific Publishing, New York, USA, Vol.1, No.2, 2013, pp. 1-18

[4] Maurya V. N., Arora Diwinder Kaur & Maurya A.K., A survey report on nonparametric hypothesis testing including ANOVA and goodness-fit-test, Communicated for publication, 2013

[5] Maurya V. N., Inferences on operating characteristics of the queue in power supply problems, Journal of Decision and Mathematical Sciences, New Delhi, India, Vol. 11, No.1-3, 2006, pp. 105-112

[6] Maurya V. N., Inferences on operating characteristics of the system size distribution in an M/G/∞:(∞;GD) queueing system in equilibrium state, Acta Ciencia Indica Mathematics, Vol. XXXII M, No.3, 2006, pp. 1093-1100, ISSN: 0970-0455 (Citation No. 015879, Indian Science Abstract, Vol. 43, No. 16, 2007)

[7] Richard E. Neapolitan, Learning Bayesian networks, North-eastern Illinois University Chicago, Illinois, 2003