Science Journal of Mathematics & Statistics (ISSN: 2276-6324), volume 2013, Article ID sjms-105, published by Science Journal Publication

# Overview of Bayesian Network

Loc Nguyen
University of Technology, Ho Chi Minh city, Vietnam

## Contents

Abstract ........ 2
1. Introduction ........ 2
2. Advanced concepts ........ 9
2.1. Markov condition ........ 11
2.2. d-separation ........ 17
2.3. Markov equivalence ........ 27
2.4. Faithfulness condition ........ 30
2.5. Other advanced concepts ........ 35
3. Inference ........ 38
3.1. Markov condition based inference ........ 39
3.2. DAG based inference ........ 47
3.3. Optimal factoring based inference ........ 50
4. Parameter learning ........ 54
4.1. Parameter learning with binomial complete data ........ 65
4.2. Parameter learning with binomial incomplete data ........ 79
4.3. Parameter learning with multinomial complete data ........ 83
5. Structure learning ........ 88
5.1. Score-based approach ........ 88
5.2. Constraint-based approach ........ 95
6. Conclusions ........ 98
References ........ 98

# Abstract

Bayesian network is a combination of probabilistic model and graph model. It is applied widely in machine learning, data mining, diagnosis, etc. because it has a solid evidence-based inference which is familiar to human intuition. However, Bayesian network may cause confusions because there are many complicated concepts, formulas and diagrams relating to it. Such concepts should be organized and presented in such a clear manner that understanding it is easy. This is the goal of this report. The report includes 5 main sections that cover principles of Bayesian network. The section 1 is an introduction to Bayesian network giving some basic concepts. Advanced concepts are mentioned in section 2. Inference mechanism of Bayesian network is described in section 3. Parameter learning which tells us how to update parameters of Bayesian network is described in section 4. Section 5 focuses on structure learning which mentions how to build up Bayesian network. In general, three main subjects of Bayesian network are inference, parameter learning, and structure learning which are mentioned in successive sections 3, 4, and 5. Section 6 is the conclusion. Main contents of this reported are extracted from the book "Learning Bayesian Networks" by Richard E. Neapolitan (2003) and the PhD dissertation "A User Modeling for Adaptive Learning" by Loc Nguyen (2014). This report focuses on discrete Bayesian network whose nodes are discrete random variables.

**Keywords:** Bayesian network, directed acyclic graph (DAG), Bayesian parameter learning, Bayesian structure learning, d-separation, score-based approach, constraint-based approach.

# 1. Introduction

This introduction section starts with a little bit discussion of Bayesian inference which is the base of both Bayesian network and inference in Bayesian network described later. Note, main content of this reported are extracted from the book "Learning Bayesian Networks" by Richard E. Neapolitan (Neapolitan, 2003) and the PhD dissertation "A User Modeling for Adaptive Learning" by Loc Nguyen (Nguyen, 2014).

**Bayesian inference** (Wikipedia, Bayesian inference, 2006), 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 by an initial probability or prior probability. Note, in classical statistical theory, the random variable's probability is objective (physical) through trials. But, in Bayesian method, the probability of hypothesis is "personal" because its initial value is set subjectively by expert. When evidences were gathered enough, the hypothesis is considered trustworthy.

Bayesian inference is based on so-called Bayes' rule or Bayes' theorem (Wikipedia, Bayesian inference, 2006) specified in equation 1.1 as follows:

$$P(H|D) = \frac{P(D|H)P(H)}{P(D)} \tag{1.1}$$

Where,

- $H$ is probability variable denoting a hypothesis existing before evidence.
- $D$ is also probabilistic variable denoting an observed evidence. It is conventional that notations $d$, $D$ and $\mathcal{D}$ are used to denote evidence, evidences, evidence sample, data sample, sample, training data and corpus (another term for data sample). Data sample or evidence sample is defined as a set of data or a set of observations which is collected by an individual, a group of persons, a computer software or a business process, which focuses on a particular analysis purpose (Wikipedia, Sample (statistics), 2014). The term "data sample"

is derived from statistics; please read the book "Applied Statistics and Probability for Engineers" by Montgomery and Runger (Montgomery & Runger, 2003, p. 4) for more details about sample and statistics.

- *P*(*H*) is *prior probability* of hypothesis *H*. It reflects the degree of subjective belief in hypothesis *H*.
- *P*(*H*|*D*), conditional probability of *H* with given *D*, is called *posterior probability*. It tells us the changed belief in hypothesis when occurring evidence. Whether or not the hypothesis in Bayesian inference is considered trustworthy is determined based on the posterior probability. In general, posterior probability is cornerstone of Bayesian inference.
- *P*(*D*|*H*) is conditional probability of occurring evidence *D* when hypothesis *H* was given. In fact, likelihood ratio is *P*(*D*|*H*) / *P*(*D*) but *P*(*D*) is constant value. So we can consider *P*(*D*|*H*) as *likelihood function* of *H* with fixed *D*. Please pay attention to the conditional probability because it is mentioned over the whole research.
- *P*(*D*) is probability of occurring evidence *D* together all mutually exclusive cases of hypothesis. If $H$ and $D$ are discrete, then $P(D) = \sum_H P(D|H)P(H)$, otherwise $f(D) = \int f(D|H)f(H)\mathrm{d}H$ with *H* and *D* being continuous, *f* denoting probability density function (Montgomery & Runger, 2003, p. 99). Because of being sum of products of prior probability and likelihood function, *P*(*D*) is called *marginal probability*.

Note: *H*, *D* must be random variables (Montgomery & Runger, 2003, p. 53) according to theory of probability and statistics and *P*(.) denotes *random probability*.

Beside Bayes' rule, there are three other rules such as additional rule, multiplication rule and total probability rule which are relevant to conditional probability. Given two random events (or random variables) *X* and *Y*, the additional rule (Montgomery & Runger, 2003, p. 33) and multiplication rule (Montgomery & Runger, 2003, p. 44) are expressed in equations 1.2 and 1.3, respectively as follows:

$$P(X \cup Y) = P(X) + P(Y) - P(X \cap Y) \tag{1.2}$$

$$P(X \cap Y) = P(X,Y) = P(X|Y)P(Y) = P(Y|X)P(X) \tag{1.3}$$

Where notations ∪ and ∩ denote union operator and intersection operator in set theory (Wikipedia, Set (mathematics), 2014). Your attention please, when *X* and *Y* are numerical variables, notations ∪ and ∩ also denote operators "or" and "and" in theory logic (Rosen, 2012, pp. 1-12). The probability *P*(*X*, *Y*) is often known as joint probability.

If $X$ and *Y* are mutually exclusive ($X \cap Y = \emptyset$) then, $X \cup Y$ is often denoted as *X*+*Y* and we have:

$$P(X + Y) = P(X) + P(Y)$$

(Due to *P*(Ø) = 0)

*X* and *Y* are mutually independent if and only if one of three following conditions is satisfied:

$$P(X \cap Y) = P(X)P(Y)$$
$$P(X|Y) = P(X)$$
$$P(Y|X) = P(Y)$$

When *X* and *Y* are mutually independent, $X \cap Y$ are often denoted as *XY* and we have:

$$P(XY) = P(X,Y) = P(X \cap Y) = P(X)P(Y)$$

Given a complete set of mutually exclusive events $X_1$, $X_2$,…, $X_n$ such that

$$X_1 \cup X_2 \cup \ldots \cup X_n = X_1 + X_2 + \cdots + X_n = \Omega \text{ where } \Omega \text{ is probability space}$$
$$X_i \cap X_j = \emptyset, \forall i,j$$

The total probability rule (Montgomery & Runger, 2003, p. 44) is specified in equation 1.4 as follows:

$$P(Y)=P(Y|X_1)P(X_1)+P(Y|X_2)P(X_2)+\cdots+P(Y|X_n)P(X_n)=\sum_{i=1}^{n}P(Y|X_i)P(X_i) \quad (1.4)$$

Where $X_1+X_2+\cdots+X_n=\Omega$ and $X_i\cap X_j=\emptyset,\forall i,j$.

If $X$ and $Y$ are continuous variables, the total probability rule is re-written in integral form as follows:

$$P(Y)=\int_{X}P(Y|X)P(X)\mathrm{d}X \quad (1.5)$$

Note, *P*(*Y*|*X*) and *P*(*X*) are continuous functions known as probability density functions mentioned right after. Please pay attention to Bayes' rule (equation 1.1) and total probability rule (equations 1.4 and 1.5) because they are used frequently over the whole research.

**Bayesian network (BN)** (Neapolitan, 2003, p. 40) is combination of graph theory and Bayesian inference. It is a directed acyclic graph (DAG) which has a set of nodes and a set of directed arcs; please pay attention to the terms "DAG" and "BN" because they are used over the whole research. By default, directed graphs in this report are DAGs if there no additional explanation. Each node represents a random variable which can be an evidence or hypothesis in Bayesian inference. Each arc reveals the relationship among two nodes. If there is the arc from node *A* to *B*, we call "*A* causes *B*" or "*A* is parent of *B*", in other words, *B* depends conditionally on *A*. Otherwise there is no arc between *A* and *B*, it asserts a conditional independence. Note, in BN context, terms: *node and variable are the same*. BN is also called *belief network*, *causal network*, or *influence diagram*, in which a name can be specific for an application type or a purpose of explanation.

Moreover, each node has a local Conditional Probability Distribution (CPD) with attention that conditional probability distribution is often called shortly *probability distribution* or *distribution*. If variables are discrete, CPD is simplified as Conditional Probability Table (CPT). If variables are continuous, CPD is often called conditional Probability Density Function (PDF) which will be mentioned in section 4 – how to learn CPT from beta density function. PDF can be called *density function*, in brief. CPD is the general term for both CPT and PDF; there is convention that CPD, CPT and PDF indicate both probability and conditional probability. In general, each CPD, CPT or PDF specifies a random variable and is known as the *probability distribution* or *distribution* of such random variable.

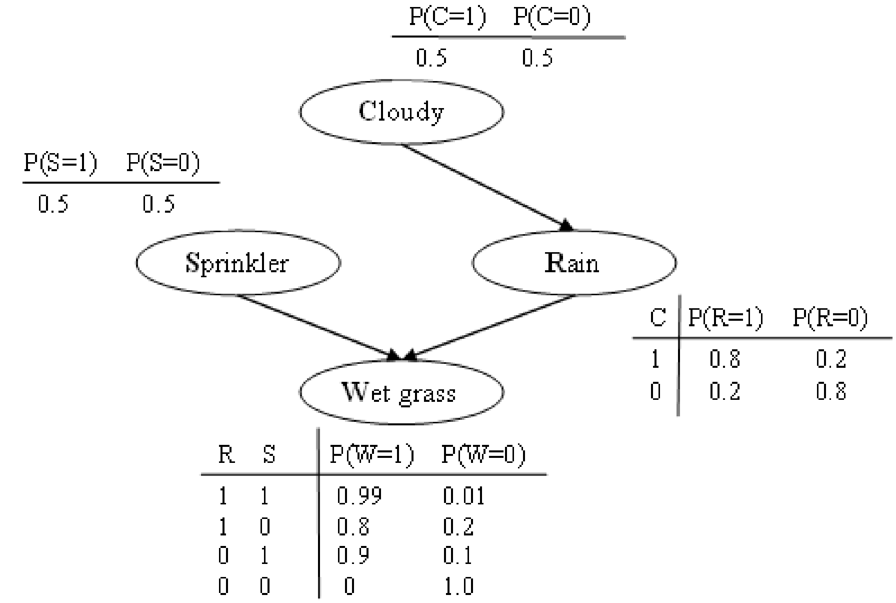
Another representation of CPD is cumulative distribution function (CDF) (Montgomery & Runger, 2003, p. 64) (Montgomery & Runger, 2003, p. 102) but CDF and PDF have the same meaning and they share interchangeable property when PDF is derivative of CDF; in other words, CDF is integral of PDF. In practical statistics, PDF is used more commonly than CDF is used and so, PDF is mentioned over the whole research. Note, notation *P*(.) often denotes probability and it can be used to denote PDF but we prefer to use lower case letters such as *f* and *g* to denote PDF. Given a variable having PDF *f*, we often state that "such variable has distribution *f* or such variable has density function *f*". Let *F*(*X*) and *f*(*X*) be CDF and PDF, respectively, equation 1.6 is the definition of CDF and PDF.

$$\text{Continuous case:}\begin{cases} F(X_0) = P(X \le X_0) = \displaystyle\int_{-\infty}^{X_0} f(X)dX \\ \displaystyle\int_{-\infty}^{+\infty} f(X)dX = 1 \end{cases}$$
$$\text{Discrete case:}\begin{cases} F(X_0) = P(X \le X_0) = \displaystyle\sum_{X \le X_0} P(X) \\ f(X) = P(X) \text{ and} \displaystyle\sum_{X} P(X) = 1 \end{cases} \tag{1.6}$$

Because this introduction section focuses on BN, please read (Montgomery & Runger, 2003, pp. 98-103) for more details about CDF and PDF.

Now please pay attention to the concept CPT because it occurs very frequently in the research; you can understand simply that CPT is essentially collection of discrete conditional probabilities of each node (variable). It is easy to infer that CPT is discrete form of PDF. When one node is conditionally dependent on another, there is a corresponding probability (in CPT or CPD) measuring the influence of causal node on this node. In case that node has no parent, its CPT *degenerates into prior probabilities*. This is the reason CPT is often identified with probabilities and conditional probabilities. This report focuses on discrete BN and so CPT is an important concept.

**Example 1.1.** In figure 1.1, event "cloudy" is cause of event "rain" which in turn is cause of "grass is wet" (Murphy, 1998). So we have three causal 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.



| P(C=1) | P(C=0) |
|---|---|
| 0.5 | 0.5 |

| P(S=1) | P(S=0) |
|---|---|
| 0.5 | 0.5 |

| C | P(R=1) | P(R=0) |
|---|---|---|
| 1 | 0.8 | 0.2 |
| 0 | 0.2 | 0.8 |

| R | S | P(W=1) | P(W=0) |
|---|---|---|---|
| 1 | 1 | 0.99 | 0.01 |
| 1 | 0 | 0.8 | 0.2 |
| 0 | 1 | 0.9 | 0.1 |
| 0 | 0 | 0 | 1.0 |

**Figure 1.1.** Bayesian network (a classic example about wet grass)

Note that random variables *C*, *S*, *R*, and *W* denote phenomena or events such as cloudy, sprinkler, rain, and wet grass, respectively and the table next to each node expresses the CPT of such node. For instance, focusing on the CPT attached to node "Wet grass", if it is rainy ($R$=1) and garden is sprinkled ($S$=1), it is almost certain that grass is wet ($W$=1). Such assertion can be represented mathematically by the condition probability of event "grass is wet" ($W$=1) given evident events "rain" ($R$=1) and "sprinkler" ($S$=1) is 0.99 as in the attached table, $P(W=1|R=1, S=1) = 0.99$. As seen, the conditional probability $P(W=1|R=1,S=1)$ is an entry of the CPT attached to node "Wet grass"■

In general, BN consists of two models such as qualitative model and quantitative model. Qualitative model is the structure as the DAG shown in figure 1.1. Quantitative model includes parameters which are CPTs attached to nodes in BN. Thus, CPTs as well as conditional probabilities are known as parameters of BN. Parameter learning and structure learning will be mentioned in sections 4 and 5. Beside important subjects of BN such as parameter learning and structure learning, there is a more essential subject which is inference mechanism inside BN when the inference mechanism is a very powerful mathematical tool that BN provides us. Before studying inference mechanism in this wet grass example, we should know other basic concepts of Bayesian network.

Let $\{X_1, X_2,\ldots, X_n\}$ be the set of nodes in BN, the joint probability distribution is defined as the probability function of event $\{X_1=x_1, X_2=x_2,\ldots, X_n=x_n\}$ (Neapolitan, 2003, p. 24). Such joint probability distribution satisfies two conditions specified by equation 1.7:

$$0 \leq P(X_1, X_2, \ldots, X_n) \leq 1$$
$$\sum_{X_1, X_2, \ldots, X_n} P(X_1, X_2, \ldots, X_n) = 1 \quad (1.7)$$

Later, we will know that a BN is modeled as the pair (*G*, *P*) where *G* is a DAG and *P* is a joint probability distribution. However, it is not easy to determine *P* by equation 1.7. As usual, *P* is defined based on Markov condition. Let $PA_i$ be the set of direct parent nodes of $X_i$. Informally, a BN satisfies Markov condition if each $X_i$ is only dependent on $PA_i$. Markov condition will be made clear in section 2.Hence, the joint probability distribution $P(X_1, X_2,\ldots, X_n)$ is defined as product of all CPTs of nodes according to equation 1.8 so that Markov condition is satisfied.

$$P(X_1, X_2, \ldots, X_n) \equiv \prod_{i=1}^{n} P(X_i|PA_i) \quad (1.8)$$

Note that $P(X_i|PA_i)$ in equation 1.8 is CPT of $X_i$ and so the joint probability distribution *P* in equation 1.8 is defined as product of CPTs.

According to Bayesian rule, given evidence (random variables) $\mathcal{D}$, the posterior probability $P(X_i|\mathcal{D})$ of variable $X_i$ is computed in equation 1.9 as below:

$$P(X_i|\mathcal{D}) = \frac{P(\mathcal{D}|X_i)P(X_i)}{P(\mathcal{D})} = \frac{P(X_i, D)}{P(\mathcal{D})} \quad (1.9)$$

Where $P(X_i)$ is prior probability of random variable $X_i$ and $P(\mathcal{D}|X_i)$ is conditional probability of occurring $\mathcal{D}$ given $X_i$ and $P(\mathcal{D})$ is probability of occurring $\mathcal{D}$ together all mutually exclusive cases of *X*. From equations 1.8 and 1.9, we gain equation 1.10 as follows:

$$P(X_i|\mathcal{D}) = \frac{P(X_i, D)}{P(\mathcal{D})} = \frac{\sum_{X\backslash(\{X_i\}\cup\mathcal{D})} P(X_1, X_2, \ldots, X_n)}{\sum_{X\backslash\mathcal{D}} P(X_1, X_2, \ldots, X_n)} \quad (1.10)$$

Where $X\backslash(\{X_i\} \cup \mathcal{D})$ and $X\backslash\mathcal{D}$ are all possible values $X = (X_1, X_2,\ldots, X_n)$ with fixing (excluding) $\{X_i\} \cup \mathcal{D}$ and fixing (excluding) $\mathcal{D}$, respectively. Note that evidence $\mathcal{D}$ including at least one random variable $X_i$ is a subset of $X$ and the sign "\" denotes the subtraction (excluding) in set theory (Wikipedia, Set (mathematics), 2014). Please pay attention that the equation 1.10 is the base for inference inside Bayesian network, which is used over the whole research. Equations 1.9 and 1.10 are extensions of Bayes' rule specified by equation 1.1. It is not easy to understand equation 1.10 and so, please see equations 1.12 and 1.13 which are advanced posterior probabilities applied into wet grass example in order to comprehend equation 1.10.

**Example 1.2.** From figure 1.1 of wet grass example, we have the joint probability $P(C, R, S, W)$ as follows:

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

Applying equation 1.8, $P(S|C)=P(S)$ due to no 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, applying equation 1.8, we have equation 1.11 for determining global joint probability distribution of "wet grass" Bayesian network as follows:

$$P(C,R,S,W) = P(C) * P(S) * P(R|C) * P(W|R,S) \tag{1.11}$$

Using Bayesian inference, 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 BN. Reviewing figure 1.1, suppose $W$ becomes evidence variable which is observed as 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 $R$ (=1) and $S$ (=1) in condition $W$ (=1). Such probabilities called *explanations* for $W$ are simple forms of equation 1.10, expended by equations 1.12 and 1.13 as follows:

$$P(R=1|W=1) = \frac{\sum_{\{C,R,S,W\}\backslash\{R=1,W=1\}} P(C,R,S,W)}{\sum_{\{C,R,S,W\}\backslash\{W=1\}} P(C,R,S,W)} = \frac{\sum_{C,S} P(C,R=1,S,W=1)}{\sum_{C,R,S} P(C,R,S,W=1)} \tag{1.12}$$

$$P(S=1|W=1) = \frac{\sum_{\{C,R,S,W\}\backslash\{S=1,W=1\}} P(C,R,S,W)}{\sum_{\{C,R,S,W\}\backslash\{W=1\}} P(C,R,S,W)} = \frac{\sum_{C,R} P(C,R,S=1,W=1)}{\sum_{C,R,S} P(C,R,S,W=1)} \tag{1.13}$$

Note that the numerator in the right side of equation 1.12 is the sum of possible probabilities $P(C, R = 1, S, W = 1)$ over possible values of $C$ and $S$. Concretely, we have an interpretation for the numerator as follows:

$$\begin{aligned}&\sum_{C,S} P(C,R=1,S,W=1)\\ &\quad = P(C=1,R=1,S=1,W=1) + P(C=1,R=1,S=0,W=1)\\ &\quad + P(C=0,R=1,S=1,W=1) + P(C=0,R=1,S=0,W=1)\end{aligned}$$

Applying equation 1.11 for global joint probability distribution of "wet grass" Bayesian network, we have:

$$\begin{aligned}&\sum_{C,S} P(C,R=1,S,W=1)\\ &= \big(P(C=1) * P(S=1) * P(R=1|C=1) * P(W=1|R=1,S=1)\big)\\ &+\big(P(C=1) * P(S=0) * P(R=1|C=1) * P(W=1|R=1,S=0)\big)\\ &+\big(P(C=0) * P(S=1) * P(R=1|C=0) * P(W=1|R=1,S=1)\big)\\ &+\big(P(C=0) * P(S=0) * P(R=1|C=0) * P(W=1|R=1,S=0)\big)\end{aligned}$$

$$= (0.5 * 0.5 * 0.8 * 0.99) + (0.5 * 0.5 * 0.8 * 0.8) + (0.5 * 0.5 * 0.2 * 0.99)$$
$$+ (0.5 * 0.5 * 0.2 * 0.8)$$
$$= 0.4475$$

It is easy to infer that there is the same interpretation for numerators and denominators in right sides of equations 1.12 and 1.13 and the previous equation 1.10 is also understood simply by this way when {*C*, *S*} = {*C*, *R*, *S*, *W*}\{*R*, *W*} and fixing {*R*, *W*}. In similar, we have:

$$\sum_{C,R} P(C, R, S = 1, W = 1)$$
$$= \big(P(C = 1) * P(S = 1) * P(R = 1|C = 1) * P(W = 1|R = 1, S = 1)\big)$$
$$+\big(P(C = 1) * P(S = 1) * P(R = 1|C = 1) * P(W = 1|R = 0, S = 1)\big)$$
$$+\big(P(C = 0) * P(S = 1) * P(R = 1|C = 0) * P(W = 1|R = 1, S = 1)\big)$$
$$+\big(P(C = 0) * P(S = 1) * P(R = 1|C = 0) * P(W = 1|R = 0, S = 1)\big)$$
$$= (0.5 * 0.5 * 0.8 * 0.99) + (0.5 * 0.5 * 0.8 * 0.9) + (0.5 * 0.5 * 0.2 * 0.99)$$
$$+ (0.5 * 0.5 * 0.2 * 0.9)$$
$$= 0.4725$$

$$\sum_{C,R,S} P(C, R, S, W = 1)$$
$$= \big(P(C = 1) * P(S = 1) * P(R = 1|C = 1) * P(W = 1|R = 1, S = 1)\big)$$
$$+\big(P(C = 1) * P(S = 0) * P(R = 1|C = 1) * P(W = 1|R = 1, S = 0)\big)$$
$$+\big(P(C = 1) * P(S = 1) * P(R = 0|C = 1) * P(W = 1|R = 0, S = 1)\big)$$
$$+\big(P(C = 1) * P(S = 0) * P(R = 0|C = 1) * P(W = 1|R = 0, S = 0)\big)$$
$$+\big(P(C = 0) * P(S = 1) * P(R = 1|C = 0) * P(W = 1|R = 1, S = 1)\big)$$
$$+\big(P(C = 0) * P(S = 0) * P(R = 1|C = 0) * P(W = 1|R = 1, S = 0)\big)$$
$$+\big(P(C = 0) * P(S = 1) * P(R = 0|C = 0) * P(W = 1|R = 0, S = 1)\big)$$
$$+\big(P(C = 0) * P(S = 0) * P(R = 0|C = 0) * P(W = 1|R = 0, S = 0)\big)$$
$$= (0.5 * 0.5 * 0.8 * 0.99) + (0.5 * 0.5 * 0.8 * 0.8) + (0.5 * 0.5 * 0.2 * 0.9)$$
$$+ (0.5 * 0.5 * 0.2 * 0) + (0.5 * 0.5 * 0.2 * 0.99) + (0.5 * 0.5 * 0.2 * 0.8)$$
$$+ (0.5 * 0.5 * 0.8 * 0.9) + (0.5 * 0.5 * 0.8 * 0)$$
$$= 0.6725$$

In fact, equations 1.12 and 1.13 are expansions of equation 1.10. As a result, we have:

$$P(R = 1|W = 1) = \frac{\sum_{C,S} P(C, R = 1, S, W = 1)}{\sum_{C,R,S} P(C, R, S, W = 1)} = \frac{0.4475}{0.6725} \approx 0.67$$
$$P(S = 1|W = 1) = \frac{\sum_{C,R} P(C, R, S = 1, W = 1)}{\sum_{C,R,S} P(C, R, S, W = 1)} = \frac{0.4725}{0.6725} \approx 0.70$$

Obviously, the posterior probability of event "sprinkler" (*S*=1) is larger than the posterior probability of event "rain" (*R*=1) given evidence "wet grass" (*W*=1), which leads to conclusion that sprinkler is the most likely cause of wet grass■

Now a short description of Bayesian is introduced. Next section will concern advanced concepts of Bayesian network.

## 2. Advanced concepts

Recall that the structure of a Bayesian network (BN) is directed acyclic graph (DAG) (Neapolitan, 2003, p. 40) in which the nodes (vertices) are linked together by directed edges (arcs); each edge expresses dependence relationships between nodes. If there is the edge from node *X* to *Y*, we call "*X* causes *Y*" or "*X* is parent of *Y*", in other words, *Y* depends conditionally on *X*. So, the edge $X \rightarrow Y$ denotes parent-child, prerequisite, or cause-effect relationship (causal relationship). Otherwise there is no edge between *X* and *Y*, it asserts the conditional independence. When we focus on cause-effect relationship in which *X* is direct cause of *Y*, the edge $X \rightarrow Y$ is called causal edge and the whole BN is called causal network. Let $V = \{X_1, X_2, X_3, \ldots, X_n\}$ and *E* be a set of nodes and a set of edges, let $G = (V, E)$ denote a DAG where *V* is a set of nodes, *E* is a set of edges, and there is no directed cycle in *G*. The "wet grass" graph shown in figure 1.1 is a DAG. Figure 2.1 (Neapolitan, 2003, p. 72) shows three DAGs.

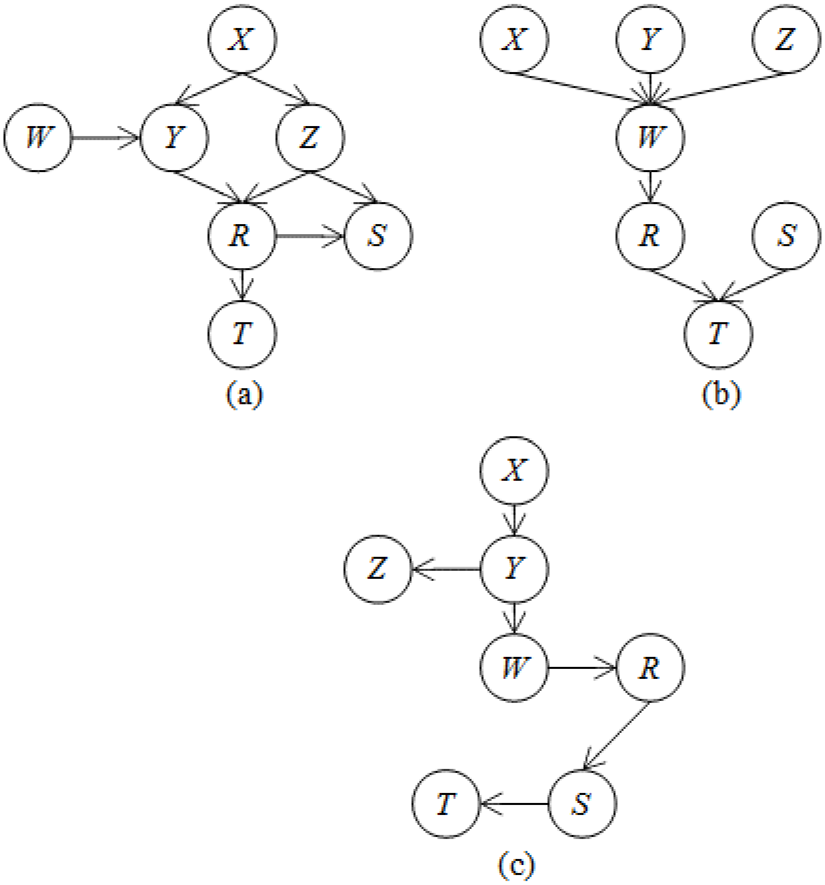


**Figure 2.1.** Three DAGs

Note that node $X_i$ is also random variable. In this report, uppercase letters, for example *X*, *Y*, *Z*, often denote random variables or set of random variables whereas lowercase letters, for example *x*, *y*, *z*, often denote their instantiations. We should glance over other popular concepts (Neapolitan, 2003, p. 31), (Neapolitan, 2003, p. 71).

- 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). Given the edge $X \rightarrow Y$, the tail is at *X* and the head is at *Y*.
- 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. A *sub-path* $X_m \rightarrow \ldots X_n$ is the path from $X_m$ to $X_n$: $X_m \rightarrow X_{m+1} \rightarrow \ldots \rightarrow X_n$ where $1 \leq m < n \leq k$. A *directed cycle* is the path from a node to itself. A *simple path* is the path that has no directed cycle. A DAG is the directed graph that has no directed cycle. By default, directed graphs in this report are DAGs if there no additional explanation. Figures 1.1 and 2.1 are examples of DAG. When we focus on cause-effect relationship in which every edge is causal edge, the DAG is called *causal DAG*.

- If there is an edge from $X$ to $Y$ then, $X$ is called *parent* of $Y$. 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 $X$ then, $Y$ 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 $X$–$Y$. Chain is denoted $X$–$Y$–$Z$, for example. A *cycle* is the chain from a node to itself. A *simple chain* is the chain that has no cycle. The concepts "adjacent" and "incident" are kept intact with link.
- A DAG $G$ is a *directed tree* if every node except root has only one parent. A DAG $G$ is called *singly-connected* if there is only one chain (if exists) between two nodes. Of course, directed tree is singly-connected DAG. In figure 2.1, the DAG (b) is a singly-connected DAG and the DAG (c) is a directed tree.

The strength of dependence between two nodes is quantified by conditional probability table (CPT) in discrete case. 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 of $X_k$ which is the conditional probability $P(X_k \mid X_i, X_j)$ has eight entries:

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

The joint probability distribution of whole BN is established according to equation 1.7.

$$0 \leq P(X_1, X_2, \ldots, X_n) \leq 1$$

$$\sum_{X_1, X_2, \ldots, X_n} P(X_1, X_2, \ldots, X_n) = 1$$

However, as usual, the joint probability distribution is formulated as product of CPTs or CPDs of nodes according to equation 1.8 so that Markov condition is satisfied, as follows:

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|PA_i)$$

Markov condition will be mentioned later. Note, the conditional probability $P(X_i|PA_i)$ is CPT of node $X_i$ where $PA_i$ is the set of direct parents of $X_i$. Let $(G, P)$ denote a BN where $G = (V, E)$ is a DAG and $P$ is a joint probability distribution. Hence, BN is a combination of probabilistic model and graph model. Note, by default, $G$ is a DAG.

Suppose a BN has $n$ binary nodes, the joint probability distribution $P(X_1, X_2, \ldots, X_n)$ requires $2^n$ entries. There is a restrictive criterion called Markov condition that makes relationships (also CPT) among nodes simpler. Firstly, we need to know concept of conditional independence and then Markov condition will be mention later. Given a DAG $G = (V, E)$, a joint probability distribution $P$, and three subsets of $V$ such as $A$, $B$, and $C$, we define:

- The denotation $I_P(A, B)$ indicates that *A* and *B* are independent (Neapolitan, 2003, p. 18), which means that $P(A, B) = P(A)P(B)$. Note, the *direct independence* $I_P(A, B)$ here is defined based on the joint probability distribution.
- The denotation $I_P(A, B \mid C)$ indicates that *A* and *B* are conditionally independent given *C* (Neapolitan, 2003, p. 19), which means that $P(A, B \mid C) = P(A \mid C)P(B \mid C)$. Note, the *conditional independence* $I_P(A, B \mid C)$ here is defined based on the joint probability distribution. The conditional independence $I_P(A, B \mid C)$ is the most general case because *C* can be empty such that $I_P(A, B \mid \emptyset) = I_P(A, B)$.

In general, equation 2.1 specified the conditional independence $I_P(A, B \mid C)$.

$$\begin{aligned}
&I_P(A,B|C) \Leftrightarrow P(A,B|C) = P(A|C)P(B|C) \\
&I_P(A,B|C) \Leftrightarrow P(A|B,C) = P(A|C) \qquad (2.1) \\
&I_P(A,B|C) \Leftrightarrow P(B|A,C) = P(B|C)
\end{aligned}$$

For convention, let $NI_P(A, B \mid C)$ denote *conditional dependence*, which means than *A* and *B* are conditionally dependent given *C*. *C* can be empty and of course we have $NI_P(A, B \mid \emptyset) = NI_P(A, B)$. Note, $NI_P(A, B)$ is also called *direct dependence* and $NI_P(A, B \mid C)$ is the inverse of $I_P(A, B \mid C)$.

$$\begin{aligned}
&NI_P(A,B|C) = \text{Not}\big(I_P(A,B|C)\big) \\
&NI_P(A,B|C) \Leftrightarrow P(A,B|C) \neq P(A|C)P(B|C) \\
&NI_P(A,B|C) \Leftrightarrow P(A|B,C) \neq P(A|C) \\
&NI_P(A,B|C) \Leftrightarrow P(B|A,C) \neq P(B|C)
\end{aligned}$$

According to definition 2.1 (Neapolitan, 2003, p. 75), two conditional independences $I_P(A_1, B_1 \mid C_1)$ and $I_P(A_2, B_2 \mid C_2)$ are *equivalent* if for every joint probability distribution *P* of *V*, $I_P(A_1, B_1 \mid C_1)$ holds if and only if $I_P(A_2, B_2 \mid C_2)$ holds. Note, *V* is the set of random variables (nodes) in $G = (V, E)$.

## 2.1. Markov condition

Recall that let (*G*, *P*) denote a BN where $G = (V, E)$ is a DAG and *P* is a joint probability distribution. **Markov condition** (Neapolitan, 2003, p. 31) is stated that every node *X* is conditionally independent from its non-descendants given its parents. In other words, node *X* is only dependent on its directed parents. Equation 2.1.1 defines Markov condition (Neapolitan, 2003, p. 31).

$$\forall X \in V, I_P(\{X\}, ND_X|PA_X) \qquad (2.1.1)$$

Where $ND_X$ and $PA_X$ are the set of non-descendants of *X* and the set of parents of *X*, respectively. As a convention, $ND_X$ excludes *X* and $PA_X$ excludes *X* too, such that $X \notin ND_X, X \notin PA_X$. $ND_X$ is not empty but $PA_X$ can be empty. When $PA_X$ is empty, equation 2.1.1 becomes:

$$\forall X \in V, I_P(\{X\}, ND_X)$$

Note, Markov condition is defined based on the joint probability distribution *P* and so, equation 2.1.1 is interpreted as follows:

$$\forall X \in V, P(\{X\}, ND_X|PA_X) = P(\{X\}|PA_X)P(ND_X|PA_X)$$

When $PA_X$ is empty,

$$\forall X \in V, P(\{X\}, ND_X) = P(\{X\})P(ND_X)$$

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

**Example 2.1.1.** Given two DAGs $G_1$ and $G_2$ shown in figure 2.1.1 and a joint probability distribution $P(X, Y, Z)$, we will test whether ($G_1$, *P*) and ($G_2$, *P*) satisfy Markov condition.

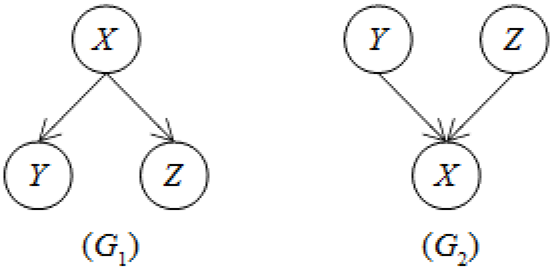


**Figure 2.1.1.** An example of two DAGs

Variable *X*, *Y*, and *Z* represents colored objects, numbered objects, and square-round objects, respectively (Neapolitan, 2003, p. 11). There are such 13 objects shown in figure 2.2.2 (Neapolitan, 2003, p. 12).

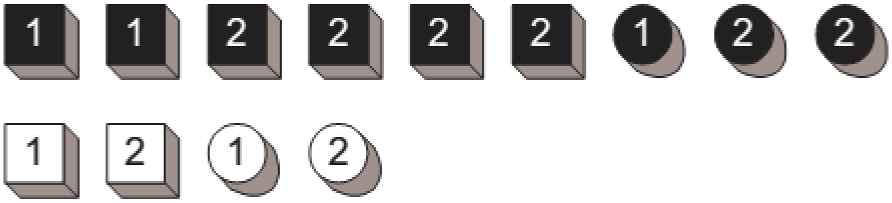


**Figure 2.1.2.** Thirteen objects

Values of *X*, *Y*, and *Z* are defined in table 2.1.1 (Neapolitan, 2003, p. 32).

| | |
|---|---|
| $X$=1 | All black objects |
| $X$=0 | All white objects |
| $Y$=1 | All object named "1" |
| $Y$=0 | All object named "2" |
| $Z$=1 | All square objects |
| $Z$=0 | All round objects |

**Table 2.1.1.** Values of variables representing thirteen objects

The joint probability distribution $P(X, Y, Z)$ assigns a probability of 1/13 to each object. In other words, $P(X, Y, Z)$ is determined as relative frequencies among such 13 objects. For example, $P(X=1, Y=1, Z=1)$ is probability of objects which are black, named "1", and square. There are 2 such objects and hence, $P(X=1, Y=1, Z=1) = 2/13$. As another example, we need to calculate the marginal probability $P(X=1, Y=1)$ and the conditional probability $P(Y=1, Z=1 \mid X=1)$. Because there are 3 black and named "1" objects, we have $P(X=1, Y=1) = 3/13$. Because there are 2 named "1" and square objects among objects 9 black objects, we have $P(Y=1, Z=1 \mid X=1) = 2/9$. It is easy to verify that the joint probability distribution $P(X, Y, Z)$ satisfies equation 1.7, as seen in table 2.1.2:

| $X, Y, Z$ | $P(X, Y, Z)$ |
|---|---|
| 1, 1, 1 | 2/13 |
| 1, 1, 0 | 1/13 |
| 1, 0, 1 | 4/13 |
| 1, 0, 0 | 2/13 |
| 0, 1, 1 | 1/13 |
| 0, 1, 0 | 1/13 |
| 0, 0, 1 | 1/13 |
| 0, 0, 0 | 1/13 |

**Table 2.1.2.** Joint probability distribution $P(X, Y, Z)$

For $(G_1, P)$, we only test whether $I_P(\{Y\}, \{Z\} \mid \{X\})$ holds because there is only one possible $I_P(\{Y\}, \{Z\} \mid \{X\})$ in $G_1$ according to Markov condition. In other words, we will test if $P(Y, Z \mid X) = P(Y \mid$

*X*)*P*(*Z* | *X*) for all values of *X*, *Y*, and *Z*. Table 2.1.3 compares *P*(*Y*, *Z* | *X*) with *P*(*Y* | *X*)*P*(*Z* | *X*) for all values of *X*, *Y*, *Z*.

| *X*, *Y*, *Z* | *P*(*Y*, *Z*\|*X*) | *P*(*Y*\|*X*)*P*(*Z*\|*X*) |
|---|---:|---|
| 1, 1, 1 | 2/9 | (3/9)*(6/9)=2/9 |
| 1, 1, 0 | 1/9 | (3/9)*(3/9)=1/9 |
| 1, 0, 1 | 4/9 | (6/9)*(6/9)=4/9 |
| 1, 0, 0 | 2/9 | (6/9)*(3/9)=2/9 |
| 0, 1, 1 | 1/4 | (2/4)*(2/4)=1/4 |
| 0, 1, 0 | 1/4 | (2/4)*(2/4)=1/4 |
| 0, 0, 1 | 1/4 | (2/4)*(2/4)=1/4 |
| 0, 0, 0 | 1/4 | (2/4)*(2/4)=1/4 |

**Table 2.1.3.** Comparison of *P*(*Y*, *Z* | *X*) with *P*(*Y* | *X*)*P*(*Z* | *X*)

From table 2.1.3, *P*(*Y*, *Z* | *X*) equals *P*(*Y* | *X*)*P*(*Z* | *X*) for all values of *X*, *Y*, and *Z*, which implies $I_P$({*Y*}, {*Z*} | {*X*}) holds. Hence, ($G_1$, *P*) satisfies Markov condition.

For ($G_2$, *P*), we only test whether $I_P$({*Y*}, {*Z*}) holds because there is only one possible $I_P$({*Y*}, {*Z*}) in $G_2$. In other words, we will test if *P*(*Y*, *Z*) = *P*(*Y*)*P*(*Z*) for all values of *Y* and *Z*. Table 2.1.4 compares *P*(*Y*, *Z*) with *P*(*Y*)*P*(*Z*) for all values of *X*, *Y*, *Z*.

| *Y*, *Z* | *P*(*Y*, *Z*) | *P*(*Y*)*P*(*Z*) |
|---|---:|---|
| 1, 1 | 3/13 | (5/13)*(8/13)=40/169 |
| 1, 0 | 2/13 | (5/13)*(5/13)=25/169 |
| 0, 1 | 5/13 | (8/13)*(8/13)=64/169 |
| 0, 0 | 3/13 | (8/13)*(5/13)=40/169 |

**Table 2.1.4.** Comparison of *P*(*Y*, *Z*) with *P*(*Y*)*P*(*Z*)

From table 2.1.4, *P*(*Y*, *Z*) is different from *P*(*Y*)*P*(*Z*) for all values of *Y* and *Z*, which implies $I_P$({*Y*}, {*Z*}) does not hold. Hence, ($G_2$, *P*) does not satisfy Markov condition■

According to theorem 2.1.1 (Neapolitan, 2003, p. 34), if Markov condition is satisfied, evaluation of the joint probability distribution equals evaluation of the product of conditions probabilities of nodes given values of their parents (Neapolitan, 2003, p. 34) whenever these conditional probabilities exist. Note, nodes are evaluated as values. For example, given $P_1$ is a joint probability distribution. Suppose we do not know the formula of $P_1$ but if (*G*, $P_1$) where *G* is a DAG satisfies Markov condition in evaluation then:

$$\forall x_i, \forall pa_i, P_1(X_1 = x_1, X_2 = x_1, \dots, X_n = x_n) = \prod_{i=1}^{n} P(X_i = x_i | PA_i = pa_i) \qquad (2.1.2)$$

The expression $\prod_{i=1}^{n} P(X_i|PA_i)$ is called *Markov condition formula*. In other words, according to theorem 2.1.1 (Neapolitan, 2003, p. 34), if a (*G*, *P*) satisfies Markov condition then *P* also satisfies Markov condition formula specified equation 2.1.2 in evaluation. Recall that the conditional probability *P*($X_i$|$PA_i$) is CPT or CPD of $X_i$. The proof of theorem 2.1.1 is in (Neapolitan, 2003, pp. 34-35).

Conversely, according to theorem 2.1.2 (Neapolitan, 2003, p. 37), given a DAG *G* and every node $X_i$ in *G* has a condition probability *P*($X_i$ | $PA_i$) on its parents. If the joint probability distribution is defined as product of conditions probabilities of nodes given their parents, $P(X_1, X_2, \dots, X_n) \equiv \prod_{i=1}^{n} P(X_i|PA_i)$ according to equation 1.8, then (*G*, *P*) satisfies Markov condition. In other words, if the joint probability distribution is defined as Markov condition formula then, the (*G*, *P*) satisfies Markov condition. The proof of theorem 2.1.1 is in (Neapolitan, 2003, pp. 37-38). Theorems 2.1.1 and 2.1.2 are corner stone of Bayesian network, which are invented by Neapolitan.

BN is constructed in practice with theorem 2.1.2 (Neapolitan, 2003, p. 37). Markov condition reduces significantly computational cost. Suppose a DAG $G$ has $n$ binary nodes, the joint probability distribution $P(X_1, X_2,\ldots, X_n)$ requires $2^n$ entries. However, given $P$ is established according to theorem 2.1.2 (Neapolitan, 2003, p. 37), if every node has at most $k$ (<<$n$) parents then, $P$ needs only $n2^k$ (<<$2^n$) entries at most because each node needs $2^k$ entries at most for its CPT.

**Example 2.1.2.** For illustrating theorem 2.1.1 (Neapolitan, 2003, p. 34), given the DAG $G_1$ shown in figure 2.1.1 and a joint probability distribution $P(X, Y, Z)$ defined as relative frequencies among 13 objects shown in figure 2.1.2. In other words, $P(X, Y, Z)$ assigns a probability of 1/13 to each object. For example, because there are 2 such objects and hence, we have $P(X=1, Y=1, Z=1)$ = 2/13. Because there are 3 black and named “1” objects, we have $P(X=1, Y=1)$ = 3/13.

From example 2.1.1, we know that ($G_1$, $P$) satisfies Markov condition according to equation 2.1.1, we will prove that the joint probability distribution $P(X, Y, Z)$ also satisfies Markov condition formula according to equation 2.1.2. The Markov condition formula for $G_1$ is $P(Y, Z|X)P(X)$.

| $X, Y, Z$ | $P(X, Y, Z)$ | $P(Y, Z\|X)P(X)$ |
|---|---:|---|
| 1, 1, 1 | 2/13 | (2/9)*(9/13)=2/13 |
| 1, 1, 0 | 1/13 | (1/9)*(9/13)=1/13 |
| 1, 0, 1 | 4/13 | (4/9)*(9/13)=4/13 |
| 1, 0, 0 | 2/13 | (2/9)*(9/13)=2/13 |
| 0, 1, 1 | 1/13 | (1/4)*(4/13)=1/13 |
| 0, 1, 0 | 1/13 | (1/4)*(4/13)=1/13 |
| 0, 0, 1 | 1/13 | (1/4)*(4/13)=1/13 |
| 0, 0, 0 | 1/13 | (1/4)*(4/13)=1/13 |

**Table 2.1.5.** Comparison of the joint probability distribution $P(X, Y, Z)$ with the Markov condition formula $P(Y, Z|X)P(X)$

From table 2.1.5, $P(X, Y, Z)$ equals $P(Y, Z|X)P(X)$ for all values of $X$, $Y$, and $Z$, which implies the joint probability distribution $P(X, Y, Z)$ also satisfies Markov condition formula according to equation 2.1.2.

For illustrating theorem 2.1.2 (Neapolitan, 2003, p. 37), from example 2.1.1, given ($G_2$, $P$) shown in figure 2.1.1 where its joint probability distribution $P(X, Y, Z)$ is defined as relative frequencies among 13 objects and its DAG $G_2$ does not satisfy Markov condition, we prove that ($G_2$, $P$) will satisfies Markov condition if $P$ is re-defined as Markov condition formula according to theorem 2.1.2 (Neapolitan, 2003, p. 37).

$$P(X,Y,Z) = P(Y)P(Z)P(X|Y,Z)$$

Note, $P(Y)$, $P(Z)$, and $P(X|Y, Z)$ are CPTs calculated as relative frequencies among 13 objects shown in figure 2.1.2. Instead of evaluating the new $P$ for all values of $X$, $Y$, and $Z$ as usual, we will prove by symbolic inference. In fact, we have:

$$P(Y=y,Z=z) = \sum_X P(X,Y=y,Z=z) = \sum_X P(Y=y)P(Z=z)P(X|Y=y,Z=z)$$

$$= P(Y=y)P(Z=z)\sum_X P(X|Y=y,Z=z) = P(Y=y)P(Z=z)$$

It implies $P(Y, Z) = P(Y)P(Z)$ for all values of $Y$ and $Z$, which means that the unique conditional independence $I_P(\{Y\}, \{Z\})$ holds. Hence $(G_2, P)$ with new $P$ satisfies Markov condition according to the definition of Markov condition specified equation 2.1.1■

Every joint probability distribution $P$ owns "inherent" conditional independences. When a $(G, P)$ satisfies Markov condition, each "Markov" conditional independence of each node from its non-descendants given its parents belongs to "inherent" conditional independences of $P$ via equation 2.1.1. In other words, that $(G, P)$ satisfies Markov condition means $G$ entails only a subset or whole of "inherent" conditional independences of $P$. For example, given $(G_1, P)$ specified by figure 2.1.1 and table 2.1.1, the $I_P(\{Y\}, \{Z\} \mid \{X\})$ is a "Markov" conditional independence of $Y$ (and $Z$) given parent node $X$ and it is also a "inherent" conditional independence derived from $P$. There is a question: whether Markov condition entails other conditional independences different from "Markov" conditional independences of nodes? Neapolitan (Neapolitan, 2003, p. 66) said yes.

According to definition 2.1.1 (Neapolitan, 2003, p. 66), let $G = (V, E)$ be a DAG, where $V$ is a set of random variables. We say that, based on the Markov condition, $G$ entails conditional independence $I_P(A, B \mid C)$ for $A, B, C \subseteq V$ if $I_P(A, B \mid C)$ holds for every $P \in \mathbb{P}$ where $\mathbb{P}$ is the set of all joint probabilities that $(G, P)$ satisfies Markov condition. Neapolitan (Neapolitan, 2003, p. 66) also said Markov condition entails the conditional independence $I_P(A, B \mid C)$ for $G$ and that the conditional independence $I_P(A, B \mid C)$ is in $G$. As a convention, such $I_P(A, B \mid C)$ is called *entailed conditional independence*. Of course, "Markov" conditional independence is an entailed conditional independence. An "inherent" conditional independence (in a $P$) that is not entailed by Markov condition is called *non-entailed conditional independence*. In general, within Markov condition, let $M$ be the set of "Markov" conditional independences, let $E$ be the set of entailed conditional independence, and let $N_P$ be the set of "inherent" conditional independences in a given $P$, we have:

$$M \subseteq E \subseteq N_P$$

Your attention please, the sets $M$ and $E$ are determined over all $P \in \mathbb{P}$ where $\mathbb{P}$ is the set of all joint probabilities that $(G, P)$ satisfies Markov condition. In other words, $M$ is the same for all $P \in \mathbb{P}$ and $E$ is the same for all $P \in \mathbb{P}$.

According to lemma 2.1.1 (Neapolitan, 2003, p. 75), any conditional independence entailed by a DAG, based on the Markov condition, is equivalent to a conditional independence among disjoint sets of random variables. Please see the aforementioned definition 2.1.1 (Neapolitan, 2003, p. 75) for more details about equivalent independence. For instance, given three sets of random variables $A$, $B$, and $C$ such that $A \cap B = \emptyset$, $A \cap C \neq \emptyset$, and $B \cap C \neq \emptyset$, that is, for every probability distribution $P$ of $V$, $I_P(A, B \mid C)$ holds if and only $I_P(A \backslash C, B \backslash C \mid C)$ holds. Obviously, $A \backslash C$, $B \backslash C$, and $C$ are disjoint sets. Note, the sign "\" denotes the subtraction (excluding) in set theory (Wikipedia, Set (mathematics), 2014).

**Example 2.1.2.** For illustrating concept of entailed conditional independence, given a DAG $G = (V, E)$ shown in figure 2.1.3 (Neapolitan, 2003, p. 67). Let $\mathbb{P}$ be the set of all joint probability distributions such that $(G, P)$ satisfies Markov condition for all $P \in \mathbb{P}$.

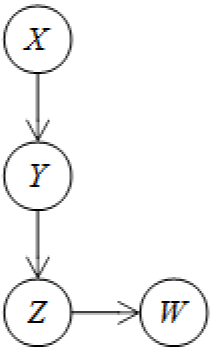


**Figure 2.1.3.** A DAG for illustrating concept of entailed conditional independence

Because the DAG in figure 2.1.3 has only two "Markov" conditional independences $I_P(\{W\}, \{X, Y\} \mid \{Z\})$ and $I_P(\{Z\}, \{X\} \mid \{Y\})$, all $P \in \mathbb{P}$ own the twos. Hence, if another conditional independence is derived from the twos, it is an entailed conditional independence entailed by Markov condition.

From $I_P(\{W\}, \{X, Y\} \mid \{Z\})$, according to equation 2.1, we have:

$$P(W|Z,X,Y) = P(W|Z,\{X,Y\}) = P(W|Z)$$

From $I_P(\{W\}, \{X, Y\} \mid \{Z\})$, according to equation 2.1, we also have:

$$P(W,\{X,Y\}|Z) = P(W,X,Y|Z) = P(W|Z)P(X,Y|Z)$$

It implies

$$P(W,Y|Z) = \sum_X P(W,X,Y|Z) = \sum_X P(W|Z)P(X,Y|Z) = P(W|Z)\sum_X P(X,Y|Z)$$
$$= P(W|Z)P(X|Z)$$

Hence, $W$ is conditionally independent from $Y$ given $Z$. In other words, we have:

$$P(W|Y,Z) = P(W|Z) = P(W|Z,X,Y)$$

From $I_P(\{Z\}, \{X\} \mid \{Y\})$ we have:

$$P(Z|X,Y) = P(Z|Y)$$

Neapolitan (Neapolitan, 2003, p. 68) proved that:

$$P(W|X,Y) = \sum_Z P(W|Z,X,Y)P(Z|X,Y)$$

(Due to total probability rule)

$$= \sum_Z P(W|Y,Z)P(Z|Y)$$

(Due to $P(W|Y, Z) = P(W|Z, X, Y)$ and $P(Z|X, Y) = P(Z|Y)$)

$$= P(W|Y)$$

(Due to total probability rule)

Obviously, $W$ and $X$ are conditionally independent given $Y$ and so it is asserted that $I_P(\{W\}, \{X\} \mid \{Y\})$ is entailed from Markov condition■

Although Markov condition entails independence, it does not entail dependence. Concretely (Neapolitan, 2003, p. 65), given a ($G$, $P$) satisfies Markov condition, the absence of an edge from node $X$ to node $Y$ implies independence of $Y$ from $X$ but the presence of an edge from node $X$ to node $Y$ does not implies dependence of $X$ and $Y$. The faithfulness condition mentioned in subsection 2.4 will matches independence and dependence with absence and presence of edges.

## 2.2. d-separation

Independence in a (*G*, *P*) until now is defined based on the joint probability distribution. For instance, given a DAG *G* = (*V*, *E*), a joint probability distribution *P*, and subsets of *V* such as *A*, *B*, and *C*, a conditional independence $I_P(A, B \mid C)$ is defined as follows:

$$I_P(A,B|C) \Leftrightarrow P(A,B|C) = P(A|C)P(B|C)$$

However, independence in a (*G*, *P*) can be defined by topology of the DAG *G* = (*V*, *E*). The concept of d-separation is used to determine topological independence. There are some important concepts that constitute the d-separation concept (Neapolitan, 2003, p. 71):

- The chain $X \rightarrow Z \rightarrow Y$ or $X \leftarrow Z \leftarrow Y$ is called *head-to-tail meeting*, in which the edges meet head-to-tail at *Z* and *Z* is a head-to-tail node on the chain. It is also called serial path.
- The chain $X \leftarrow Z \rightarrow Y$ is called *tail-to-tail meeting*, in which the edges meet tail-to-tail at *Z* and *Z* is a tail-to-tail node on the chain. It is also called divergent chain.
- The chain $X \rightarrow Z \leftarrow Y$ is called *head-to-head meeting*, in which the edges meet head-to-head at *Z*, and *Z* is a head-to-head node on the chain. It is also called convergent chain.
- The chain *X*–*Z*–*Y* is called *uncoupled meeting* if *X* and *Y* aren't adjacent.

Let *X*, *Y* be two nodes and let *C* be a subset of nodes such that $C \subseteq V$, $X \in V\backslash C$, $Y \in V\backslash C$, and $X \neq Y$. Note, *C* can be empty. Given the chain *p* between *X* and *Y*, *p* is *blocked* by *C* if and only if one of three following *blocked conditions* is satisfied (Neapolitan, 2003, pp. 71-72):

1. There is an intermediate node $Z \in C$ on *p* so that all edges on *p* incident to *Z* are head-to-tail meeting at *Z*.
2. There is an intermediate node $Z \in C$ on *p* so that all edges on *p* incident to *Z* are tail-to-tail meeting at *Z*.
3. There is an intermediate node *Z* on *p* so that:
    - *Z* and all descendants of *Z* are not in *C* ($\notin C$).
    - All edges op *p* incident to *Z* are head-to-head meetings.

The chain is called *active* given set *C* if it is not blocked by set *C*. The third blocked condition implies that all head-to-head meetings on *p* are outside *C*. *When C is empty (C = Ø), only the third block condition is tested for blocking* because obviously the first and second blocked conditions are not satisfied with empty *C*.

**Example 2.2.1.** The DAG shown in figure 2.2.1 is used for illustrating blocked conditions.

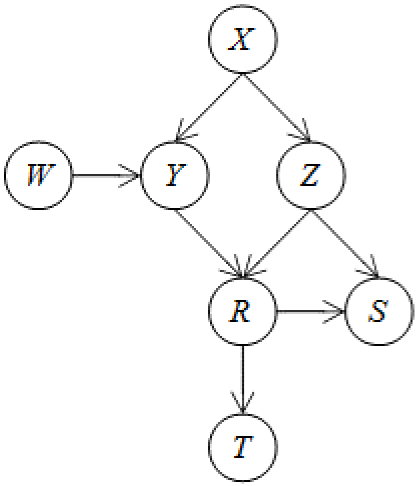


**Figure 2.2.1.** A DAG for illustrating blocked conditions

In figure 2.2.1, we have:

- The chain *Y*–*X*–*Z*–*S* is blocked by $\{X\}$ because the edges on the chain incident to *X* meet tail-to-tail at *X*, according to the condition 1 (Neapolitan, 2003, p. 72). That chain is also blocked by $\{Z\}$ because the edges on the chain incident to *Z* meet head-to-tail at *Z*, according to the condition 1 (Neapolitan, 2003, p. 72).
- The chain *W*–*Y*–*R*–*Z*–*S* is blocked by $\emptyset$ because $R \notin \emptyset$, $T \notin \emptyset$, and the edges on the chain incident to *R* (such as *Y*→*R* and *Z*→*R*) meet head-to-head at *R*, according to the condition 3 (Neapolitan, 2003, p. 72).
- The chain *W*–*Y*–*R*–*S* is blocked by $\{R\}$ because the edges on the chain incident to *R* meet head-to-tail at *R*, according to condition 1 (Neapolitan, 2003, p. 72).
- The chain *W*–*Y*–*R*–*Z*–*S* is not blocked by $\{R\}$ because the edges on the chain incident to *R* (such as *Y*→*R* and *Z*→*R*) meet head-to-head at *R*. Moreover, this chain is not blocked by $\{T\}$ because *T* is a descendent of *R* (Neapolitan, 2003, p. 72)■

According to definition 2.2.1 (Neapolitan, 2003, p. 72), given a DAG *G* = (*E*, *V*), a subset *C* ⊆ *V*, and two nodes *X* and *Y* are distinct and not in *C*. We say *X* and *Y* are **d-separated** by *C* if all chains between *X* and *Y* are blocked by *C*. *C* is also called a *d-separating* of *G*.

**Example 2.2.2.** In figure 2.2.1, we have:

- *X* and *R* are d-separated by $\{Y, Z\}$ because the chain *X*–*Y*–*R* is blocked at *Y*, and the chain *X*–*Z*–*R* is blocked at *Z* (Neapolitan, 2003, p. 72).
- *X* and *T* are d-separated by $\{Y, Z\}$ because the chain *X*–*Y*–*R*–*T* is blocked at *Y*, the chain *X*–*Z*–*R*–*T* is blocked at *Z*, and the chain *X*–*Z*–*S*–*R*–*T* is blocked at *Z* and at *S* (Neapolitan, 2003, p. 72).
- *Y* and *Z* are d-separated by $\{X\}$ because the chain *Y*–*X*–*Z* is blocked at *X*, the chain *Y*–*R*–*Z* is head-to-head meeting at *R* whereas *R* along with its descendants $\{S, T\}$ are not in $\{X\}$, and the chain *Y*–*R*–*S*–*Z* is head-to-head meeting at *S* whereas *S* is not in $\{X\}$ (Neapolitan, 2003, p. 72).
- *W* and *S* are d-separated by $\{R, Z\}$ because the chain *W*–*Y*–*R*–*S* is blocked at *R*, the chains *W*–*Y*–*R*–*Z*–*S* and *W*–*Y*–*X*–*Z*–*S* are both blocked at *Z* (Neapolitan, 2003, p. 72).
- *W* and *S* are also d-separated by $\{Y, Z\}$ because the chain *W*–*Y*–*R*–*S* is blocked at *Y*, the chain *W*–*Y*–*R*–*Z*–*S* is blocked at $\{Y, Z\}$, and the chain *W*–*Y*–*X*–*Z*–*S* is blocked at *Z* (Neapolitan, 2003, p. 72). The chain *W*–*Y*–*R*–*Z*–*S* is also head-to-head meeting at *R* whereas *R* along with its descendants $\{S, T\}$ are not in $\{Y, Z\}$.
- *W* and *T* are d-separated by $\{R\}$ because the chains *W*–*Y*–*R*–*T*, *W*–*Y*–*X*–*Z*–*R*–*T*, and *W*–*Y*–*X*–*Z*–*S*–*R*–*T* are blocked at *R*.
- *W* and *X* are not d-separated by $\{Y\}$ because there is a chain *W*–*Y*–*X* between *W* and *Y* which is not blocked at *Y* (Neapolitan, 2003, p. 72).
- *W* and *T* are not d-separated by $\{Y\}$ because there is a chain *W*–*Y*–*X*–*Z*–*R*–*T* between *W* and *T* which is not blocked at *Y* (Neapolitan, 2003, p. 72). Note, none of three blocked conditions for $\{Y\}$ is satisfied on the chain *W*–*Y*–*X*–*Z*–*R*–*T*■

According to definition 2.2.1 (Neapolitan, 2003, p. 73), given DAG *G* = (*V*, *E*) and given *A*, *B*, and *C* are mutually disjoint subsets of *V*, if for every node $X \in A$ and every node $Y \in B$, *X* and *Y* are d-separated by *C* then, we have a topological independence denoted as follows:

$$I_G(A, B|C)$$

If *C* is empty, we write:

$$I_G(A, B)$$

Hence, $I_G(A, B \mid C)$ denotes a *d-separation* where $C$ can be empty. The notation $NI_G(A, B \mid C)$ indicates that $A$ and $B$ are not d-separated by $C$ and $C$ can be empty. $NI_G(A, B \mid C)$ representing a topological dependence is the inverse of $I_G(A, B \mid C)$.

$$NI_G(A, B|C) = \text{Not}\big(I_G(A, B|C)\big)$$

Of course, we have $I_G(A, B \mid \varnothing) = I_G(A, B)$ and $NI_G(A, B \mid \varnothing) = NI_G(A, B)$.

According to lemma 2.2.1 (Neapolitan, 2003, p. 85), let $G = (V, E)$ be a DAG then, node $X$ and node $Y$ are adjacent in $G$ if and only if they are not d-separated by some set in $G$. According to corollary 2.2.1 in (Neapolitan, 2003, p. 86), let $G = (V, E)$ be a DAG then, if node $X$ and node $Y$ are d-separated by some set, they are d-separated either by the set consisting of the parents of $X$ or the set consisting of the parents of $Y$. According to lemma 2.2.2 (Neapolitan, 2003, p. 86), given a DAG $G = (V, E)$ and an uncoupled meeting $X$–$Z$–$Y$, the three following statements are equivalent:

1. $X$–$Z$–$Y$ is a head-to-head meeting.
2. There exists a set not containing $Z$ that d-separates $X$ and $Y$.
3. All sets containing $Z$ do not d-separate $X$ and $Y$.

Lemma 2.2.3 (Neapolitan, 2003, p. 74) is used to link conditional independence (probabilistic independence) and topological independence (d-separation). According to this lemma, let $G = (V, E)$ be a DAG and let $P$ be a joint probability distribution, the $(G, P)$ satisfies Markov condition if and only if

$$\forall A, B, C \subseteq V, I_G(A, B|C) \Rightarrow I_P(A, B|C) \qquad (2.2.1)$$

Where $A$, $B$, and $C$ are mutually disjoint subsets of $V$. From lemma 2.2.3 (Neapolitan, 2003, p. 74), when the $(G, P)$ satisfies Markov condition, the DAG $G$ is called an *independence map* of $P$.

**Example 2.2.3.** Given a $(G, P)$ satisfies Markov condition where $G$ is the DAG shown in figure 2.2.1 and $P$ is a joint probability distribution. We have $I_G(\{X\}, \{R\} \mid \{Y, Z\})$ because the chain $X$–$Y$–$R$ is blocked at $Y$, and the chain $X$–$Z$–$R$ is blocked at $Z$ (Neapolitan, 2003, p. 72). Because $(G, P)$ satisfies Markov condition, we also have $I_P(\{X\}, \{R\} \mid \{Y, Z\})$ according to lemma 2.2.3 (Neapolitan, 2003, p. 74). Indeed, we have:

$$P(R, X|\{Y, Z\}) = \sum_W P(R, X, W|\{Y, Z\}) = \sum_W P(R, \{X, W\}|\{Y, Z\})$$

(Due to total probability rule)

$$= \sum_W P(R|\{Y, Z\})P(\{X, W\}|\{Y, Z\})$$

(Due to Markov condition for node $R$)

$$= P(R|\{Y, Z\}) \sum_W P(\{X, W\}|\{Y, Z\})$$

$$= P(R|\{Y, Z\}) \sum_W P(X, W|\{Y, Z\}) = P(R|\{Y, Z\})P(X|\{Y, Z\})$$

(Due to total probability rule)

Obviously, we have $I_P(\{X\}, \{R\} \mid \{Y, Z\})$.

Lemma 2.2.3 (Neapolitan, 2003, p. 74) implies that, based on Markov condition, given a DAG, every d-separation is a conditional independence. Conversely, given a $(G, P)$ satisfies Markov condition, it is not sure that a conditional independence $I_P(A, B \mid C)$ in $P$ implies a d-separation $I_G(A, B \mid C)$ as seen in equation 2.2.1. This one-way rule causes a so-called explaining away phenomenon (Fenton, Noguchi, & Neil, 2019) or Berkson's paradox. Recall that there are three meetings mentioned in blocked conditions: head-to-tail (serial), tail-to-tail (divergent), and head-to-head (convergent). Three DAGs in figure 2.2.2 represent such three meetings. For extension, node *Z* in (a), (b), and (c) can be replaced by a set.

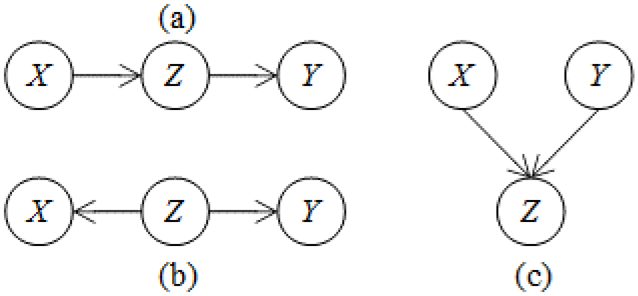


**Figure 2.2.2.** Head-to-tail (a), tail-to-tail (b), and head-to-head (c)

*X* and *Y* are not d-separated on chains (a) and (b) by Ø because three blocked conditions are not satisfied here without intermediate nodes (set of immediate nodes is empty). So, we have $NI_G(\{X\}, \{Y\})$ on chains (a) and (b). However, *X* and *Y* are d-separated on chains (a) and (b) by *Z* if *Z* is instantiated (*Z* is known) according to the first and second blocked conditions. So, we have $I_G(\{X\}, \{Y\} \mid \{Z\})$ on chains (a) and (b).

Conversely, *X* and *Y* are d-separated on chain (c) by Ø according to the third blocked condition. So, we have $I_G(\{X\}, \{Y\})$ on chain (c). However, *X* and *Y* are not d-separated on chain (c) by *Z* if *Z* is instantiated (*Z* is known) because three blocked conditions are not satisfied here by the intermediate node *Z*. So, we have $NI_G(\{X\}, \{Y\} \mid \{Z\})$ on chains (c). The existence of both $I_G(\{X\}, \{Y\})$ and $NI_G(\{X\}, \{Y\} \mid \{Z\})$ on chain (c) is the explaining away phenomenon or Berkson's paradox because we often expect that *X* is independent from *Y* given *Z* if we knew that *X* and *Y* are independent each other before. The explaining away phenomenon is illustrated in example 2.2.4. It is interesting that known *Z* blocks chains (a) and (b) at *Z* by $I_G(\{X\}, \{Y\} \mid \{Z\})$ but opens chain (c) at *Z* by $NI_G(\{X\}, \{Y\} \mid \{Z\})$.

**Example 2.2.4.** For illustrating the explaining away phenomenon, let (*G*, *P*) satisfies Markov condition where DAG *G* is shown in figure 2.2.2 (c) and *P* is a join probability distribution. From the d-separation $I_G(\{X\}, \{Y\})$, we have $I_P(\{X\}, \{Y\})$ according to lemma 2.2.3 (Neapolitan, 2003, p. 74). Suppose both *X* and *Y* are failure causes of an engine *Z*. Engine is failed when *Z*=1 (*Z* is known). If we continue to know that *X* (*Y*) is the real failure cause, the probability of *Y* (*X*) is decreased, following $NI_G(\{X\}, \{Y\} \mid \{Z\})$. This means that if *Z* is known, *X* and *Y* influence each other. Suppose failure causes have the same possibility at original state (engine *Z* is not failed yet) and so we have: $P(X=1) = P(X=0) = P(Y=1) = P(Y=0) = 0.5$, and $P(Z=1|X=1, Y=0) = P(Z=1|X=0, Y=1) = 0.8$. Followings are pre-defined CPTs of *X*, *Y*, and *Z*.

$$P(X=1) = P(X=0) = 0.5$$
$$P(Y=1) = P(Y=0) = 0.5$$
$$P(Z=1|X=1, Y=0) = P(Z=1|X=0, Y=1) = 0.8$$
$$P(Z=0|X=1, Y=0) = P(Z=0|X=0, Y=1) = 0.2$$
$$P(Z=1|X=1, Y=1) = 0.9$$
$$P(Z=0|X=1, Y=1) = 0.1$$
$$P(Z=1|X=0, Y=0) = 0$$
$$P(Z=0|X=0, Y=0) = 1$$

Due to "Markov" conditional independence $I_P(\{X\}, \{Y\})$, we have:

$$P(X, Y) = P(X)P(Y),\ P(Y|X) = P(Y), \text{ and } P(X|Y) = P(X)$$

Suppose engine *Z* is failed (*Z*=1) and we know *X* is the real failure cause (*X*=1), we need to calculate and compare $P(Y=1|X=1, Z=1)$ with $P(Y=1|Z=1)$. We have:

$$P(Z=1|X=1)=\sum_{Y}P(Z=1|X=1,Y)P(Y|X=1)=\sum_{Y}P(Z=1|X=1,Y)P(Y)$$
$$=P(Z=1|X=1,Y=1)P(Y=1)+P(Z=1|X=1,Y=0)P(Y=0)$$
$$=0.9*0.5+0.8*0.5=0.85$$

We also have:

$$P(Z=1|Y=1)=\sum_{X}P(Z=1|X,Y=1)P(X|Y=1)=\sum_{X}P(Z=1|X,Y=1)P(X)$$
$$=P(Z=1|X=1,Y=1)P(X=1)+P(Z=1|X=0,Y=1)P(X=0)$$
$$=0.9*0.5+0.8*0.5=0.85$$

Hence,

$$P(Z=1)=\sum_{X,Y}P(Z=1|X,Y)P(X,Y)=\sum_{X,Y}P(Z=1|X,Y)P(X)P(Y)$$
$$=P(Z=1|X=1,Y=1)P(X=1)P(Y=1)$$
$$+P(Z=1|X=1,Y=0)P(X=1)P(Y=0)$$
$$+P(Z=1|X=0,Y=1)P(X=0)P(Y=1)$$
$$+P(Z=1|X=0,Y=0)P(X=0)P(Y=0)$$
$$=0.9*0.5*0.5+0.8*0.5*0.5+0.8*0.5*0.5+0*0*0.5$$
$$=0.625$$

Hence,

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

(Due to Bayes' rule)

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

(Due to *I<sub>P</sub>*(*X*, *Y*))

$$=\frac{P(Z=1|X=1,Y=1)P(Y=1)}{P(Z=1|X=1)}=\frac{0.9*0.5}{0.85}\approx 0.53$$

Whereas,

$$P(Y=1|Z=1)=\frac{P(Z=1|Y=1)P(Y=1)}{P(Z=1)}=\frac{0.85*0.5}{0.625}=0.68$$

Obviously, that $P(Y=1|X=1, Z=1) < P(Y=1|Z=1)$ means $X$ and $Y$ are influence each other given $Z$, which indicates existence of the conditional dependence $NI_P(X, Y \mid Z)$ following $NI_G(X, Y \mid Z)$ in this example■

Recall that, lemma 2.2.3 (Neapolitan, 2003, p. 74) implies that, based on Markov condition, given a DAG, every d-separation is a conditional independence. Conversely, given a ($G$, $P$) satisfies Markov condition, it is not sure that a conditional independence $I_P(A, B \mid C)$ in $P$ implies a d-separation $I_G(A, B \mid C)$ as seen in equation 2.2.1. However, an entailed conditional independence always implies a d-separation (Neapolitan, 2003, p. 75). Lemma 2.2.4 (Neapolitan, 2003, p. 75) proved this. Recall that, according to definition 2.1.1 (Neapolitan, 2003, p. 66), Markov condition can entail (entailed) conditional independences which are different from "Markov" conditional independences.

According to lemma 2.2.4 (Neapolitan, 2003, p. 75), let $G = (V, E)$ be a DAG and $\mathbb{P}$ be the set of all probability distributions $P \in \mathbb{P}$ such that the $(G, P)$ satisfies the Markov condition. Then for every three mutually disjoint subsets $A, B, C \subseteq V$,

$$I_P(A, B|C) \text{ for all } P \in \mathbb{P} \Rightarrow I_G(A, B|C) \tag{2.2.2}$$

It is easy to recognize that every $I_P(A, B | C)$ in equation 2.2.2 is an entailed conditional independence, according to definition 2.1.1 (Neapolitan, 2003, p. 66).

According to definition 2.2.2 (Neapolitan, 2003, p. 76), a conditional independence $I_P(A, B | C)$ is *identified by d-separation* in $G$ if one of two following conditions is satisfied:

1. $I_G(A, B | C)$ holds.
2. $A$, $B$, and $C$ are not mutually disjoint; $A'$, $B'$, and $C'$ are mutually disjoint, $I_P(A, B | C)$ and $I_P(A', B' | C')$ are equivalent, and we have $I_G(A', B' | C')$.

Recall that two conditional dependences $I_P(A_1, B_1 | C_1)$ and $I_P(A_2, B_2 | C_2)$ are equivalent if for every joint probability distribution $P$ of $V$, $I_P(A_1, B_1 | C_1)$ holds if and only if $I_P(A_2, B_2 | C_2)$ holds, according to definition 2.1 (Neapolitan, 2003, p. 75).

As a result, according to theorem 2.2.1 (Neapolitan, 2003, p. 76), based on the Markov condition, a DAG $G$ entails all and only (entailed) conditional independencies that are identified by d-separations in $G$. In other words, there is no entailed conditional independence that is not identified by d-separation in a $(G, P)$ satisfying Markov condition where $G$ is DAG (Neapolitan, 2003, p. 75). However, with Markov condition, some non-entailed conditional independencies in a given $(G, P)$ may not be identified by d-separation, as seen in example 2.2.5.

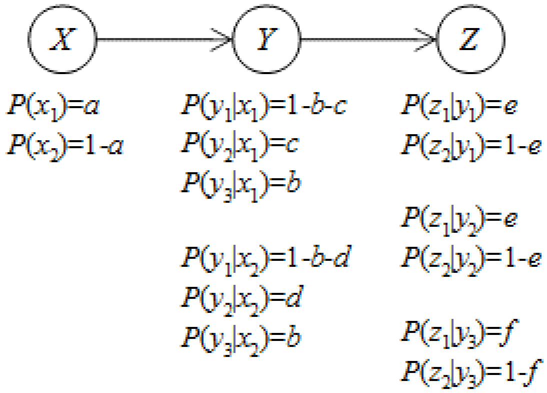


**Figure 2.2.3.** A $(G, P)$ for illustrating non-entailed conditional independence not identified by d-separation

**Example 2.2.5.** Given a $(G, P)$ shown in figure 2.2.3 (Neapolitan, 2003, p. 76), $I_G(\{X\}, \{Z\}) = I_G(\{X\}, \{Z\} | \varnothing)$ does not hold because three blocked conditions are not satisfied here without intermediate nodes (set of immediate nodes is Ø). However, $I_P(\{X\}, \{Z\})$ holds because $P(Z|X)$ equals $P(Z)$ as seen in table 2.2.1.

| $X, Z$ | $P(Z\|X)$ | $P(Z)$ |
|---|---|---|
| $x_1, z_1$ | $e - b(e - f)$ | $e - b(e - f)$ |
| $x_1, z_2$ | $1 - e + b(e - f)$ | $1 - e + b(e - f)$ |
| $x_2, z_1$ | $e - b(e - f)$ | $e - b(e - f)$ |
| $x_2, z_2$ | $1 - e + b(e - f)$ | $1 - e + b(e - f)$ |

**Table 2.2.1.** Comparison between $P(Z|X)$ and $P(Z)$ given $P$ shown in figure 2.2.3

Followings are formulas of $P(Z|X)$ and $P(Z)$.

$$P(Z|X) = \sum_Y P(Z|Y,X)P(Y|X)$$

(Due to total probability rule)

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

($P(Z|Y, X) = P(Z|Y)$ due to $I_P(X, Z \mid Y)$ according to Markov condition)

We also have:

$$P(Y)=\sum_{X} P(Y|X)P(X)$$

$$P(Z)=\sum_{Y} P(Z|Y)P(Y)$$

For explaining table 2.2.1, we try to calculate $P(Z=z_1|X=x_1)$ and $P(Z=z_1)$ as follows:

$$P(Z=z_1|X=x_1)=\sum_{Y} P(Z=z_1|Y)P(Y|X=x_1)$$

$$=P(Z=z_1|Y=y_1)P(Y=y_1|X=x_1)+P(Z=z_1|Y=y_2)P(Y=y_2|X=x_1)$$
$$+P(Z=z_1|Y=y_3)P(Y=y_3|X=x_1)$$
$$=e(1-b-c)+ec+fb=e-b(e-f)$$

$$P(Y=y_1)=\sum_{X} P(Y=y_1|X)P(X)$$
$$=P(Y=y_1|X=x_1)P(X=x_1)+P(Y=y_1|X=x_2)P(X=x_2)$$
$$=(1-b-c)a+(1-b-d)(1-a)$$

$$P(Y=y_2)=\sum_{X} P(Y=y_2|X)P(X)$$
$$=P(Y=y_2|X=x_1)P(X=x_1)+P(Y=y_2|X=x_2)P(X=x_2)$$
$$=ca+d(1-a)$$

$$P(Y=y_3)=\sum_{X} P(Y=y_3|X)P(X)$$
$$=P(Y=y_3|X=x_1)P(X=x_1)+P(Y=y_3|X=x_2)P(X=x_2)$$
$$=b$$

$$P(Z=z_1)=\sum_{Y} P(Z=z_1|Y)P(Y)$$
$$=P(Z=z_1|Y=y_1)P(Y=y_1)+P(Z=z_1|Y=y_2)P(Y=y_2)$$
$$+P(Z=z_1|Y=y_3)P(Y=y_3)$$
$$=e\big((1-b-c)a+(1-b-d)(1-a)\big)+e\big(ca+d(1-a)\big)+fb=e-b(e-f)$$

The conditional independence $I_P(\{X\}, \{Z\})$ is non-entailed conditional independence because there are many joint probability distributions (different from the one shown in figure 2.2.3) which satisfy Markov condition and Markov condition with these distributions does not entail $I_P(\{X\}, \{Z\})$. As a result, we have the non-entailed conditional independence $I_P(\{X\}, \{Z\})$ but do not have $I_G(\{X\}, \{Z\})$ (Neapolitan, 2003, p. 76). In other words, $I_P(\{X\}, \{Z\})$ is not identified by a respective d-separation■

Given DAG $G = (V, E)$, let $B$ and $C$ be sets of nodes such that $B \neq C$. In other words, $B$ and $C$ are disjoint subsets of $V$. The algorithm to find d-separations is essentially to find a set $A$ so that all nodes in $A$ are d-separated from all nodes in $B$ by $C$. This algorithm is called *find-d-separations algorithm*. Actually, find-d-separations algorithm is to find a set $A$ so that $A$ is d-separated from $B$ by $C$, which means that the d-separation $I_G(A, B \mid C)$ is determined. Note, $A \neq B$ and $A \neq C$. Let $R$ be another set of nodes, recall that a chain $p$ between node $X \in R$ and node $Z \in B$ is active given $C$ if it is not blocked by $C$ according to three blocked conditions aforementioned (Neapolitan, 2003, pp. 71-72). By negating the three blocked conditions, in other words, a triple active chain $p = [X, Y, Z]$ given $C$ where $X \in R$, $Z \in B$ must satisfy one of two following conditions (Neapolitan, 2003, p. 79):

1. $X–Y–Z$ is not head-to-head meeting at $Y$ and $Y$ is not in $C$.
2. $X–Y–Z$ is head-to-head meeting at $Y$ and $Y$ is or has a descendant in $C$.

The two conditions are called *active conditions* given $C$. So, find-d-separations algorithm aims to determine the set $R$ such that for each $X \in R$ then $X \in B$ or there is an active chain given $C$ between $X$ and a node in $B$. Finally, we have the result $A = V\backslash(C \cup R)$ where the sign "\" denotes the subtraction (excluding) in set theory (Wikipedia, Set (mathematics), 2014). The two active conditions are used to determine all active chains given $C$ here with note that an active chain is combination of successive triple active chains.

We define that an ordered pair of links ($X–Y$, $Y–Z$) in $G$ is *legal* if $X–Y–Z$ is a triple active chain which satisfies one of two active conditions, given $C$. A chain is legal if it does not contain any illegal ordered pair of links. As a convention, any link $X–Y$ is legal chain. Given $X \in B$, a node $Z$ is called reachable node of $X$ if there is a legal chain between $X$ and $Z$ with note that $X$ is considered as reachable node of $X$. A so-called *find-reachable-nodes algorithm* is to find reachable nodes of the set $B$. This implies that find-reachable-nodes algorithm is to determine the set $R$ because $R$ is essentially the set of reachable nodes of the set $B$. Obviously, find-d-separations algorithm is based on find-reachable-nodes algorithm because the aimed result is $A = V\backslash(C \cup R)$. For illustration, given $X \in B$, find-reachable-nodes algorithm find all reachable nodes of $X$ as follows (Neapolitan, 2003, p. 77): for any node $Y$ such that link $X–Y$ exists, we label the link $X–Y$ with $l$=1 and add $X$ to $R$. Next for each such $Y$, we check all unlabeled links $Y–Z$. If the pair ($X–Y$, $Y–Z$) is legal, we label the link $Y–Z$ with $l$=2 and then add $Y$ and $Z$ to $R$. We repeat this procedure with $Y$ taking the place of $X$, $Z$ taking the place of $Y$, and label $l$=3. If no more legal pair is found, the algorithm is stopped. The algorithm is similar to breadth-first graph search algorithm except that we visit links instead of visiting nodes (Neapolitan, 2003, p. 77). Note, the algorithm does not assume $G$ is DAG. Following is pseudo-code of find-reachable-nodes algorithm (Neapolitan, 2003, p. 78).

```
Inputs: a DAG $G = (V, E)$, a subset $B \subset V$
Outputs: the subset $R \subset V$ of all nodes reachable from $B$.
void find-reachable-nodes ($G = (V, E)$, set-of-nodes $B$, set-of-nodes& $R$)
{
    for (each $X \in B$) {
        add $X$ to $R$;
        for (each $Y$ such that the link $X–Y$ exists) {
            add $Y$ to $R$;
            label $X–Y$ with 1;
        }
    }
    $i$ = 1;
    found = true;
    while (found) {
        found = false;
        for (each Y such that X–Y is labeled i)
            for (each unlabeled link Y–Z
            such that (X–Y, Y–Z) is legal) {
                add Z to R;
                label Y–Z with i + 1;
                found = true;
            }
            i = i + 1;
        }
    }
}■
```

**Example 2.2.5.** Given the graph *G* = (*V*, *E*) shown in figure 2.2.4, given *B*={*X*} and *C* = {*M*}, by applying find-reachable-nodes algorithm, reachable nodes of *B*={*X*} are shaded cells such as *X*, *Y*, *N*, and *Z*. Iterations are described as follows:

- Iteration 1: Unlabeled edges *X*→*Y* and *X*→*N* are labeled 1. Nodes *X*, *Y* and *N* are added to *R* and so, *R* = {*X*, *Y*, *N*}. Legal chains are *X*→*Y*, *X*→*N*.
- Iteration 2: Unlabeled edge *Y*→*Z* are labeled 2 because the pair (*X*–*Y*, *Y*–*Z*) is legal according to the first active condition. Node *Z* is added to *R* and so, *R* = {*X*, *Y*, *N*, *Z*}. Legal chains are *X*→*Y*→*Z*, *X*→*N*.
- Iteration 3: Unlabeled edge *Z*→*N* are labeled 2 because the pair (*X*–*N*, *N*–*Z*) is legal according to the second active condition. Legal chains are *X*→*Y*→*Z*, *X*→*N*←*Z*. The algorithm is stopped because there is no more legal pair■

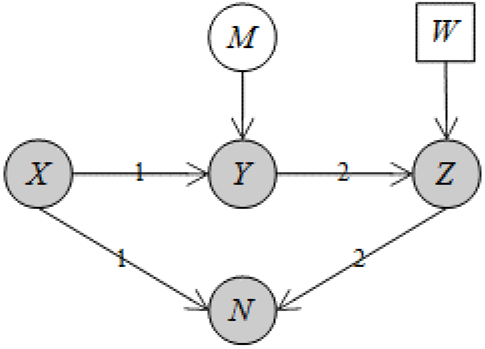


**Figure 2.2.4.** Illustrated graph *G* = (*V*, *E*) for find-reachable-nodes algorithm

Although find-d-separations algorithm is based on find-reachable-nodes algorithm, there is an adjustment is added to find-d-separations algorithm because find-reachable-nodes algorithm may ignore some reachable nodes of given node *X* or may ignore some legal chains. The reason is that some active chains are missed due to related edges were already labeled before (Neapolitan, 2003, p. 79). For example, in figure 2.2.4, the legal chain *X*→*Y*→*Z*→*N* is missed because the edge *Z*→*N* was already labeled when the legal chain *X*→*N*←*Z* was visited (Neapolitan, 2003, p. 79). This problem is solved by creating a new graph *G*' = (*V*, *E*') and then applying find-reachable-nodes algorithm into *G*' with an adjustment (Neapolitan, 2003, p. 79). The graph *G*' = (*V*, *E*') has the same nodes with the origin graph *G* = (*V*, *E*) but its set of edges *E*' is composed as *E*' = *E* ∪ {*X*→*Y* such that *X*←*Y* ∈ *E*}. Additional edges in *E*' are drawn as dash-line arrows in figured 2.2.5. The adjustment is that any ordered pair of links (*X*→*Y*, *Y*→*Z*) *in G'* is legal if *X*–*Y*–*Z* is a triple active chain which satisfies one of two active conditions *in G*. Following is pseudo-code of find-d-separations algorithm (Neapolitan, 2003, p. 79).

```
Inputs: a DAG $G = (V, E)$ and two disjoint subsets $B, C \subset V$.
Outputs: the subset $A \subset V$ containing all nodes d-separated from every node in $B$ by $C$.
void find-d-separations (
    $G = (V, E)$,
    set-of-nodes $B, C$,
    set-of-nodes& $A$)
{
    for (each $Y \in V$) {
        if ($Y \in B$)
            $in[Y]$ = true;
        else
            $in[Y]$ = false;
        if ($Y$ is or has a descendent in $C$)
            $descendent[Y]$ = true;
        else
            $descendent[Y]$ = false;
    }
    $E' = E \cup \{X \rightarrow Y$ such that $X \leftarrow Y \in E\}$

    // Call find-reachable-nodes algorithm follows:
    find-reachable-nodes algorithm ($G' = (V, E'), B, R$);

    // Use this rule to decide whether ($X–Y$, $Y–Z$) in $G'$ is legal in $G$:
    // The pair ($X–Y$, $Y–Z$) is legal if and only if $X \neq Z$
    // and one of the following holds:
    // 1) $X–Y–Z$ is not head-to-head in $G$ and $in[V]$ is false.
    // 2) $X–Y–Z$ is head-to-head in $G$ and $descendent[V]$ is true.

    $A = V \setminus (C \cup R)$; // We do not need to remove $B$ because $B \subseteq R$.
}■
```

**Example 2.2.6.** Given the graph $G' = (V, E')$ shown in figure 2.2.5 which created from the graph $G = (V, E)$ shown in figure 2.2.4, given $B=\{X\}$ and $C = \{M\}$, by applying find-d-separations algorithm, the set of reachable nodes is $R = \{X, Y, N, Z\}$ which is drawn as solid cells and the resulted set is $A = V \backslash (C \cup R) = \{W\}$ which is drawn as a rectangle cell. Obviously, the d-separation $I_G(A, B \mid C)$ is determined. Iterations are described as follows:

- Iteration 1: Unlabeled edges $X \rightarrow Y$ and $X \rightarrow N$ in $G'$ are labeled 1. Nodes $X$, $Y$ and $N$ are added to $R$ and so, $R = \{X, Y, N\}$. Legal chains are $X \rightarrow Y$, $X \rightarrow N$.
- Iteration 2: Unlabeled edge $Y \rightarrow Z$ in $G'$ are labeled 2 because the pair ($X–Y$, $Y–Z$) is legal in $G$ according to the first active condition. Node $Z$ is added to $R$ and so, $R = \{X, Y, N, Z\}$. Legal chains are $X \rightarrow Y \rightarrow Z$, $X \rightarrow N$.
- Iteration 3: Unlabeled edge $Z \rightarrow N$ in $G'$ are labeled 2 because the pair ($X–N$, $N–Z$) is legal in $G$ according to the second active condition. Legal chains are $X \rightarrow Y \rightarrow Z$, $X \rightarrow N \leftarrow Z$.
- Iteration 4: Unlabeled edge $Z \leftarrow N$ in $G'$ are labeled 3 because the pair ($Y–Z$, $Z–N$) is legal in $G$ according to the first active condition. Legal chains are $X \rightarrow Y \rightarrow Z \rightarrow N$, $X \rightarrow N \leftarrow Z$. The algorithm is stopped because there is no more legal pair■

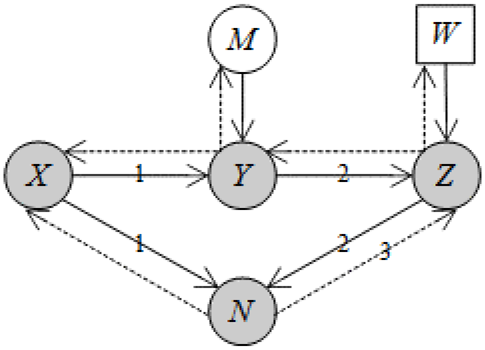


**Figure 2.2.5.** Illustrated graph $G' = (V, E')$ for find-d-separations algorithm

Theorem 2.2.2 (Neapolitan, 2003, p. 82) asserts that the resulted set $A$ returned from find-d-separations algorithm contains all and only nodes d-separated from every node in $B$ by $C$. Of course, there is no superset of such $A$.

## 2.3. Markov equivalence

DAGs which have the same set of nodes are **Markov equivalent** if and only if they have the same *d*-separations. In other words, DAGs that are Markov equivalent have the same topological independences. Equation 2.3.1 (Neapolitan, 2003, pp. 84-85) defines Markov equivalence in formal, given two DAGs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ are Markov equivalent if and only if

$$\forall A, B, C \subseteq V, I_{G_1}(A, B|C) \Leftrightarrow I_{G_2}(A, B|C) \tag{2.3.1}$$

Where $A$, $B$, and $C$ are mutually disjoint subsets of $V$. Shortly, Markov condition is defined based on joint probability distribution whereas Markov equivalence is defined based on topology of DAG (d-separation). Hence, theorem 2.3.1 and corollary 2.2.2 (Neapolitan, 2003, p. 85) are used to connect Markov condition and Markov equivalence. According to theorem 2.3.1 (Neapolitan, 2003, p. 85), two DAGs are Markov equivalent if and only if, based on the Markov condition, they entail the same (entailed) conditional independencies. According to corollary 2.2.2 (Neapolitan, 2003, p. 85), let $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ be two DAGs containing the same set of variables $V$ then, $G_1$ and $G_2$ are Markov equivalent if and only if for every probability distribution $P$ of $V$, $(G_1, P)$ satisfies the Markov condition if and only if $(G_2, P)$ satisfies the Markov condition.

According to lemma 2.3.1 (Neapolitan, 2003, p. 86), if two DAGs $G_1$ and $G_2$ are Markov equivalent, then arbitrary nodes $X$ and $Y$ are adjacent in $G_1$ if and only if they are adjacent in $G_2$. So, Markov equivalent DAGs have the same links (edges without regard for direction). According to theorem 2.3.2 (Neapolitan, 2003, p. 87), two DAGs $G_1$ and $G_2$ are Markov equivalent if and only if they have the same links (edges without regard for direction) and the same set of uncoupled head-to-head meetings. Please pay attention to theorem 2.3.2 because it is often used to check if two DAGs are Markov equivalent.

**Example 2.3.1.** Figure 2.3.1 shows four DAGs (a), (b), (c), and (d) (Neapolitan, 2003, p. 90).

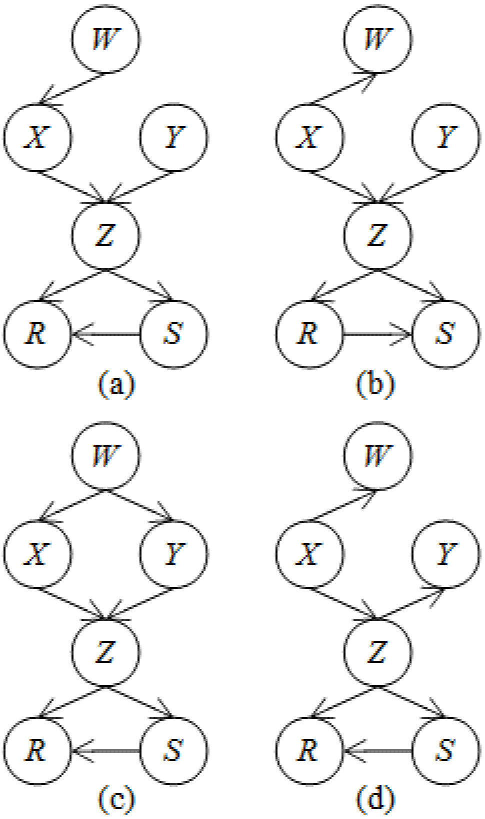


**Figure 2.3.1.** Four DAGs for illustrating Markov equivalence

According to (Neapolitan, 2003, p. 90), in figure 2.3.1, the DAGs (a) and (b) are Markov equivalent because they have the same links and have an uncoupled head-to-head meeting *X*→*Z*←*Y*. The DAG (c) is not Markov equivalent to DAGs (a) and (b) because it has the link *W*–*Y*. The DAG (d) is not Markov equivalent to DAGs (a) and (b) because although it has the same links, it does not have the uncoupled head-to-head meeting *X*→*Z*←*Y*. Of course, the DAGs (c) and (d) are not Markov equivalent to each other■

From lemma 2.3.1 and theorem 2.3.2 (Neapolitan, 2003, pp. 86-87), Neapolitan (Neapolitan, 2003, p. 91) stated that *Markov equivalence class* can be represented with a single graph that has the same links and the same uncoupled head-to-head meetings as the DAGs in the class. Note, a single graph has neither loop and nor multiple edge. Markov equivalence divides all DAGs into disjoint Markov equivalence classes. For example, figure 2.3.2 (Neapolitan, 2003, p. 85) shows three DAGs of the same Markov equivalence class and there is no other DAG which is Markov equivalent to them.

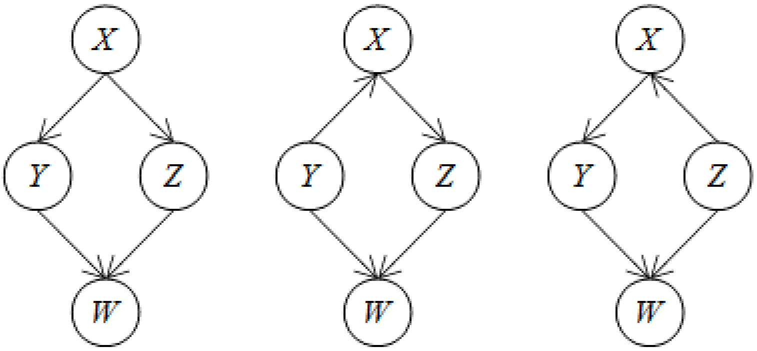


**Figure 2.3.2.** Three DAGs of the same Markov equivalence class

If we assign a direction to a link and such assignment does not produce a head-to-head meeting then, we create a new member of the existing equivalence class but we do not create a new equivalence class. For instance (Neapolitan, 2003, p. 91), if a Markov equivalence class has the edge $X \rightarrow Y$ and the uncoupled meeting $X \rightarrow Y - Z$ is not head-to-head then, all the DAGs in the equivalence class must have $Y - Z$ oriented as $Y \rightarrow Z$.

According to (Neapolitan, 2003, p. 91), a **DAG pattern** is defined for a Markov equivalence class to be the graph that has the same links as the DAGs in the equivalence class and has oriented all and only the edges common to all DAGs in the equivalence class. Edges (directed links) in DAG pattern are called *compelled edges*. In general, DAG pattern is the representation of Markov equivalence class. Figure 2.3.3 is the DAG pattern of the Markov equivalence class in figure 2.3.2.

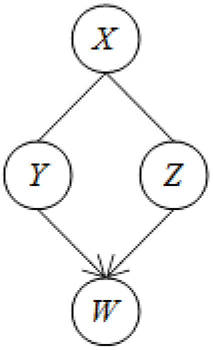


**Figure 2.3.3.** DAG pattern of the Markov equivalence class in figure 2.3

DAG pattern is the core of Bayesian structure learning. Note, DAG pattern can have both edges and links; so, DAG pattern is not a DAG and it is only a single graph. Therefore, we should survey properties of DAG pattern.

According to definition 2.3.1 (Neapolitan, 2003, p. 91), let *gp* be a DAG pattern whose nodes are the elements of *V*, and *A*, *B*, and *C* be mutually disjoint subsets of *V*. Then *A* and *B* are d-separated by *C* in *gp* if *A* and *B* are d-separated by *C* in every DAG in the Markov equivalence class represented by *gp*. This implies the DAG pattern *gp* has the same set of d-separations to all DAGs in the Markov equivalence class represented by *gp*. For example, the DAG pattern *gp* in figure 2.3.3 has the d-separation $I_{gp}(\{Y\}, \{Z\} \mid \{X\})$ because $\{Y\}$ and $\{Z\}$ are d-separated by $\{X\}$ in all DAGs shown in figure 2.3.2.

Two lemmas 2.3.2 and 2.3.3 (Neapolitan, 2003, p. 91) are derived from the definition 2.3.1 in (Neapolitan, 2003, p. 91). According to lemma 2.3.2 (Neapolitan, 2003, p. 91), let *gp* be DAG pattern and *X* and *Y* be nodes in *gp* then, *X* and *Y* are adjacent in *gp* if and only if they are not d-separated by some set in *gp*. According to lemma 2.3.3 (Neapolitan, 2003, p. 91), suppose we have a DAG pattern *gp* and an uncoupled meeting $X–Z–Y$ then, the three followings are equivalent:

1. $X–Z–Y$ is a head-to-head meeting.
2. There exists a set not containing *Z* that d-separates *X* and *Y*.
3. All sets containing *Z* do not d-separate *X* and *Y*.

Lemmas 2.3.2 and 2.3.3 are extensions of lemma 2.2.1 (Neapolitan, 2003, p. 85) and lemma 2.2.2 (Neapolitan, 2003, p. 86), respectively for DAG pattern.

Recall that when a (*G*, *P*) satisfies Markov condition, *G* is called an independence map of *P* according to lemma 2.2.3 (Neapolitan, 2003, p. 74), which causes that then every DAG which is Markov equivalent to *G* is also an independence map of *P*. As a result (Neapolitan, 2003, p. 92), based on Markov condition, DAG pattern *gp* representing the equivalence class is an independence map of *P*.

## 2.4. Faithfulness condition

From theorem 2.1.1 (Neapolitan, 2003, p. 34) and theorem 2.1.2 (Neapolitan, 2003, p. 37), Markov condition entails independence but it does not entail dependence. Lemma 2.2.1 (Neapolitan, 2003, p. 85) means that, let *G* = (*V*, *E*) be a DAG and *P* be a joint probability distribution then, the presence (absence) of edge between node *X* and node *Y* in *G* if and only if there is the absence (presence) of d-separation between *X* and *Y* by some set *C* in *G*. In successive, lemma 2.2.3 in (Neapolitan, 2003, p. 74) means that, based on Markov condition, the presence of d-separation between *X* and *Y* implies $I_P(\{X\}, \{Y\} \mid C\}$ but the absence of d-separation between *X* and *Y* does not imply $NI_P(\{X\}, \{Y\} \mid C\}$. As a result (Neapolitan, 2003, p. 65), given Markov condition, the absence of edge between *X* and *Y* implies $I_P(\{X\}, \{Y\})$ but the presence of edge between *X* and *Y* does not imply $NI_P(\{X\}, \{Y\})$.

Recall that

[

According to lemma 2.2.3 (Neapolitan, 2003, p. 74), let *G* = (*V*, *E*) be a DAG and let *P* be a joint probability distribution, the (*G*, *P*) satisfies Markov condition if and only if

$$\forall A, B, C \subseteq V, I_G(A, B|C) \Rightarrow I_P(A, B|C)$$

According to lemma 2.2.1 (Neapolitan, 2003, p. 85), let *G* = (*V*, *E*) be a DAG then, node *X* and node *Y* are adjacent in *G* if and only if they are not d-separated by some set in *G*

$$X, Y \text{ adjacent } \Leftrightarrow \exists C \subset V, NI_G(\{X\}, \{Y\}|C)$$

Note, *C* can be empty, *C* = Ø.

]

Another condition called faithfulness condition (Neapolitan, 2003, p. 65) will entail both independence and dependence between two nodes based on both absence and presence of their edge. Faithfulness condition is essential to structure learning (Neapolitan, 2003, p. 542). Before defining faithfulness condition, we need to survey some relevant concepts. A DAG is called *complete DAG* (Neapolitan, 2003, p. 94) if there always exits an edge between two arbitrary nodes. Given a complete DAG *G*, a (*G*, *P*) satisfies Markov condition for all joint probability distribution *P* because Markov condition does not entail any conditional independence in the complete DAG *G*. Two DAGs in figure 2.4.1 satisfy Markov condition for all joint probability distribution because they are complete DAGs.

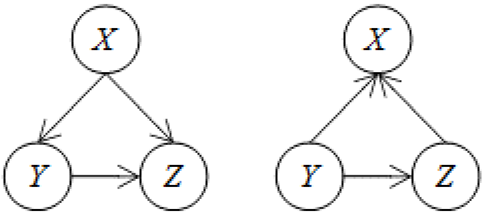


**Figure 2.4.1.** Complete DAGs

Given a probability distribution *P* and two nodes *X* and *Y*, there is a *direct dependence* between *X* and *Y* in *P* if $\{X\}$ and $\{Y\}$ are not conditionally independent (Neapolitan, 2003, p. 94) given any subset of *V* with note that Ø is also a subset of *V*. Inferring from lemma 2.2.1 (Neapolitan, 2003, p. 85), the direct dependence between *X* and *Y* implies an edge between *X* and *Y*, but why? Following is proof. Lemma 2.2.3 implies that (*G*, *P*) satisfies Markov condition if and only if

$$\forall A, B, C \subseteq V, NI_G(A, B|C) \vee I_P(A, B|C)$$

Direct dependence between *X* and *Y* in *P* means there is no conditionally independence between $\{X\}$ and $\{Y\}$ given any subset *C*.

$$\forall C \subseteq V, NI_P(\{X\}, \{Y\}|C)$$

Which implies

$$\forall C \subseteq V, NI_G(\{X\}, \{Y\}|C)$$

If there is no edge between $X$ and $Y$ then, in case of $C = \emptyset$, for an node $Z$ such that a path $p$ between $X$ and $Y$ through $Z$ must not be head-to-head meeting at $Z$ due to $NI_G(\{X\}, \{Y\})$, which lead to an event that there is a d-separation $I_G(\{X\}, \{Y\} \mid \{Z\})$. This is a contradiction from the assumption $\forall C \subseteq V, NI_G(\{X\}, \{Y\}|C)$. If such path $p$ does not exist, $X$ is totally separated from $Y$ and so this assumption is also violated. Hence, there must be an edge between $X$ and $Y$. Note, direct dependence between $X$ and $Y$ implies an edge between $X$ and $Y$ but it is not asserted in vice versa■

Inferring from both lemma 2.2.1 (Neapolitan, 2003, p. 85) and lemma 2.2.3 (Neapolitan, 2003, p. 74), given Markov condition, the absence of an edge between $X$ any $Y$ implies there is no direct dependency between $X$ and $Y$ (there exists $I_P(\{X\}, \{Y\} \mid C)$ with some $C$), but the presence of an edge between $X$ and $Y$ does not imply there is a direct dependency (there exists $NI_P(\{X\}, \{Y\} \mid C)$ for all $C$).

According to definition 2.4.1 (Neapolitan, 2003, p. 95), given a joint probability distribution $P$ and a DAG $G = (V, E)$, the $(G, P)$ satisfies **faithfulness condition** if two following conditions are satisfied:

1. $(G, P)$ satisfies Markov condition, which means that $G$ entails only ("inherent") conditional independences in $P$.
2. All conditional independences in $P$ are entailed by $G$, based on Markov condition.

In other words, a $(G, P)$ satisfies faithfulness condition if $G$ entails only and all conditional independences in $P$, based on Markov condition. It is easy to recognize that, within faithfulness condition, the set of entailed conditional independences is the set of "inherent" conditional independences in $P$. Recall that, within only Markov condition, the set of entailed conditional independences is subset of "inherent" conditional independences in $P$. So, faithfulness condition is stronger than Markov condition. When $(G, P)$ satisfies the faithfulness condition, we say $P$ and $G$ are *faithful to each other*, and we say $G$ is a *perfect map* of $P$ (Neapolitan, 2003, p. 95).

Hence, given a joint probability distribution $P$, faithfulness condition indicates that an edge between $X$ any $Y$ implies direct dependence $NI_P(\{X\}, \{Y\})$ and no edge between $X$ any $Y$ implies conditional independence $I_P(\{X\}, \{Y\})$. Note, within faithfulness condition, direct dependence between $X$ and $Y$ is the same to $NI_P(\{X\}, \{Y\})$. In general, conditional independence (probabilistic independence) is equivalent to d-separation (topological independence). As a result, let $G = (V, E)$ be a DAG and let $P$ be a joint probability distribution, the $(G, P)$ satisfies faithfulness condition if and only if

$$\forall A, B, C \subseteq V, I_G(A, B|C) \Leftrightarrow I_P(A, B|C) \tag{2.4.1}$$

Note, the sign "⇔" means "necessary and sufficient condition" or "equivalence".

**Example 2.4.1.** For illustrating faithfulness condition, given a DAG $G$ and a joint probability distribution $P(X, Y, Z)$ shown in figure 2.4.2, we will test whether $(G, P)$ satisfies faithfulness condition.

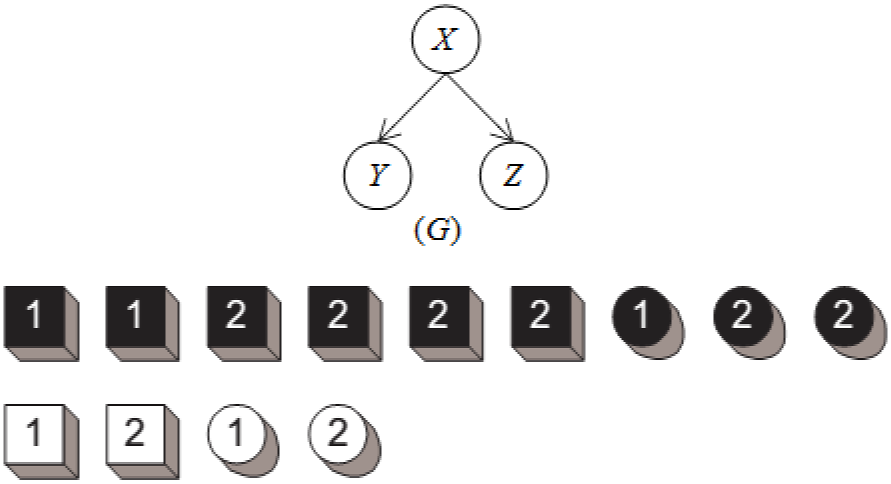


**Figure 2.4.2.** (*G*, *P*) satisfies faithfulness condition

Variable *X*, *Y*, and *Z* represents colored objects, numbered objects, and square-round objects, respectively (Neapolitan, 2003, p. 11). There are such 13 objects shown in figures 2.2.2 and 2.4.1 (Neapolitan, 2003, p. 12). Values of *X*, *Y*, and *Z* are defined as seen in table 2.1.1 (Neapolitan, 2003, p. 32):

| | |
|---|---|
| $X$=1 | All black objects |
| $X$=0 | All white objects |
| $Y$=1 | All object named "1" |
| $Y$=0 | All object named "2" |
| $Z$=1 | All square objects |
| $Z$=0 | All round objects |

The joint probability distribution *P*(*X*, *Y*, *Z*) assigns a probability of 1/13 to each object. In other words, *P*(*X*, *Y*, *Z*) is determined as relative frequencies among such 13 objects. For example, *P*(*X*=1, *Y*=1, *Z*=1) is probability of objects which are black, named "1", and square. There are 2 such objects and hence, *P*(*X*=1, *Y*=1, *Z*=1) = 2/13. As another example, we need to calculate the marginal probability *P*(*X*=1, *Y*=1) and the conditional probability *P*(*Y*=1, *Z*=1 | *X*=1). Because there are 3 black and named "1" objects, we have *P*(*X*=1, *Y*=1) = 3/13. Because there are 2 named "1" and square objects among objects 9 black objects, we have *P*(*Y*=1, *Z*=1 | *X*=1) = 2/9. It is easy to verify that the joint probability distribution *P*(*X*, *Y*, *Z*) satisfies equation 1.7, as follows:

| *X*, *Y*, *Z* | *P*(*X*, *Y*, *Z*) |
|---|---|
| 1, 1, 1 | 2/13 |
| 1, 1, 0 | 1/13 |
| 1, 0, 1 | 4/13 |
| 1, 0, 0 | 2/13 |
| 0, 1, 1 | 1/13 |
| 0, 1, 0 | 1/13 |
| 0, 0, 1 | 1/13 |
| 0, 0, 0 | 1/13 |

Hence, the (*G*, *P*) in example 2.4.1 here is as same as the ($G_1$, *P*) in example 2.1.1. There is only one "Markov" conditional independence $I_P(\{Y\}, \{Z\} \mid \{X\})$ of (*G*, *P*) but there may be six possible "inherent" conditional independences in *P* such as $I_P(\{X\}, \{Y\})$, $I_P(\{X\}, \{Z\})$, $I_P(\{Y\}, \{Z\})$, $I_P(\{X\}, \{Y\} \mid \{Z\})$, $I_P(\{X\}, \{Z\} \mid \{Y\})$, and $I_P(\{Y\}, \{Z\} \mid \{X\})$. Table 2.4.1 compares *P*(*X*, *Y*), *P*(*X*)*P*(*Y*), *P*(*X*, *Z*), *P*(*X*)*P*(*Z*), *P*(*Y*, *Z*), and *P*(*Y*)*P*(*Z*).

| $X, Y, Z$ | $P(X, Y)$ | $P(X)P(Y)$ | $P(X, Z)$ | $P(X)P(Z)$ | $P(Y, Z)$ | $P(Y)P(Z)$ |
|---|---:|---:|---:|---:|---:|---:|
| 1, 1, 1 | 3/13 | 45/169 | 6/13 | 72/169 | 3/13 | 40/169 |
| 1, 1, 0 | 3/13 | 45/169 | 3/13 | 45/169 | 2/13 | 25/169 |
| 1, 0, 1 | 6/13 | 72/169 | 6/13 | 72/169 | 5/13 | 64/169 |
| 1, 0, 0 | 6/13 | 72/169 | 3/13 | 45/169 | 3/13 | 40/169 |
| 0, 1, 1 | 2/13 | 20/169 | 2/13 | 32/169 | 3/13 | 40/169 |
| 0, 1, 0 | 2/13 | 20/169 | 2/13 | 20/169 | 2/13 | 25/169 |
| 0, 0, 1 | 2/13 | 32/169 | 2/13 | 32/169 | 5/13 | 64/169 |
| 0, 0, 0 | 2/13 | 32/169 | 2/13 | 20/169 | 3/13 | 40/169 |

**Table 2.4.1.** Comparison of $P(X, Y)$, $P(X)P(Y)$, $P(X, Z)$, $P(X)P(Z)$, $P(Y, Z)$, and $P(Y)P(Z)$

From table 2.4.1, three $I_P(\{X\}, \{Y\})$, $I_P(\{X\}, \{Z\})$, and $I_P(\{Y\}, \{Z\})$ do not hold because $P(X, Y) \neq P(X)P(Y)$, $P(X, Z) \neq P(X)P(Z)$, $P(Y, Z) \neq P(Y)P(Z)$. Table 2.4.2 compares $P(X, Y|Z)$, $P(X|Z)P(Y|Z)$, $P(X, Z|Y)$, $P(X|Y)P(Z|Y)$, $P(Y, Z|X)$, and $P(Y|X)P(Z|X)$.

| $X, Y, Z$ | $P(X, Y\|Z)$ | $P(X\|Z)P(Y\|Z)$ | $P(X, Z\|Y)$ | $P(X\|Y)P(Z\|Y)$ | $P(Y, Z\|X)$ | $P(Y\|X)P(Z\|X)$ |
|---|---:|---:|---:|---:|---:|---:|
| 1, 1, 1 | 1/4 | (6/8)*(3/8) =9/32 | 2/5 | (3/5)*(3/5) =9/25 | 2/9 | 2/9 |
| 1, 1, 0 | 1/5 | (3/5)*(2/5) =6/25 | 1/5 | (3/5)*(2/5) =6/25 | 1/9 | 1/9 |
| 1, 0, 1 | 1/2 | (6/8)*(5/8) =15/32 | 1/2 | (6/8)*(5/8) =15/32 | 4/9 | 4/9 |
| 1, 0, 0 | 2/5 | (3/5)*(3/5) =9/25 | 1/4 | (6/8)*(3/8) =9/32 | 2/9 | 2/9 |
| 0, 1, 1 | 1/8 | (2/8)*(3/8) =3/32 | 1/5 | (2/5)*(3/5) =6/25 | 1/4 | 1/4 |
| 0, 1, 0 | 1/5 | (2/5)*(2/5) =4/25 | 1/5 | (2/5)*(2/5) =4/25 | 1/4 | 1/4 |
| 0, 0, 1 | 1/8 | (2/8)*(5/8) =5/32 | 1/8 | (2/8)*(5/8) =5/32 | 1/4 | 1/4 |
| 0, 0, 0 | 1/5 | (2/5)*( 3/5) =6/25 | 1/8 | (2/8)*(3/8) =3/32 | 1/4 | 1/4 |

**Table 2.4.1.** Comparison of $P(X, Y|Z)$, $P(X|Z)P(Y|Z)$, $P(X, Z|Y)$, $P(X|Y)P(Z|Y)$, $P(Y, Z|X)$, and $P(Y|X)P(Z|X)$

From table 2.4.1, because there is only one equality $P(Y, Z|X) = P(Y|X)P(Z|X)$, only "inherent" conditional independence $I_P(\{X\}, \{Z\}\} \mid \{Y\})$ which is also the unique "Markov" conditional independence holds. This implies Markov condition entails only and all "inherent" conditional independences in $P$. Hence, according to definition 2.4.1 (Neapolitan, 2003, p. 95), $(G, P)$ satisfies faithfulness condition■

Theorem 2.4.1 (Neapolitan, 2003, p. 96) and theorem 2.4.2 (Neapolitan, 2003, p. 97) connect faithfulness condition and topological independences. According to theorem 2.4.1 (Neapolitan, 2003, p. 96), a $(G, P)$ satisfies faithfulness condition if and only if all and only conditional independencies in $P$ are identified by d-separations in the DAG $G$, as follows:

$$\forall A, B, C \subseteq V, I_G(A, B|C) \Leftrightarrow I_P(A, B|C)$$

Going back example 2.2.5, we have $I_P(\{X\}, \{Z\})$ but we do not have $I_G(\{X\}, \{Z\})$ and so the $(G, P)$ in example 2.2.5 does not satisfies faithfulness condition. Please view example 2.2.5 to know how to determine $I_P(\{X\}, \{Z\})$.

According to theorem 2.4.2 (Neapolitan, 2003, p. 97), if (*G*, *P*) satisfies faithfulness condition, then *P* satisfies this faithfulness condition with all and only DAGs that are Markov equivalent to the DAG *G*. Furthermore, if we let *gp* be the DAG pattern corresponding to this Markov equivalence class then, d-separations in *gp* identify all and only conditional independencies in *P*. In other words, all and only conditional independencies in *P* are identified by d-separations in *gp*. We say that *gp* and *P* are faithful to each other, and *gp* is a perfect map of *P*.

According to Neapolitan (Neapolitan, 2003, p. 97), we say a joint probability distribution *P* *admits a faithful DAG representation* if *P* is faithful to some DAG (and therefore some DAG pattern). It is easy to infer from theorem 2.4.2 (Neapolitan, 2003, p. 97) that if *P* admits a faithful DAG representation, there exists a unique DAG pattern with which *P* is faithful. The goal of structure learning is to find such unique DAG pattern if we knew *P* is faithful to some DAG (*P* admits a faithful DAG representation) before.

According to theorem 2.4.3 (Neapolitan, 2003, p. 99), suppose a joint probability distribution *P* admits some faithful DAG representation then, *gp* is the DAG pattern faithful to *P* if and only if the two following conditions are satisfied:

1. *X* and *Y* are adjacent in *gp* if and only if there is no subset $S \subseteq V$ such that $I_P(\{X\}, \{Y\} \mid S)$ holds. That is, *X* and *Y* are adjacent if and only if there is a direct dependence between *X* and *Y*.
2. Any chain $X-Z-Y$ is a head-to-head meeting in *gp* if and only if $Z \in S$ implies $NI_P(\{X\}, \{Y\} \mid S)$).

Following is proof of theorem 2.4.3 (Neapolitan, 2003, p. 99). From theorem 2.4.2, if the DAG pattern is faithful to *P*, all and only conditional independencies in *P* are identified by d-separations in *gp*, which means that the two conditions are satisfied when condition 1 is combination of lemma 2.2.3 (Neapolitan, 2003, p. 74) and lemma 2.2.1 (Neapolitan, 2003, p. 85) and condition 2 is combination of lemma 2.2.3 (Neapolitan, 2003, p. 74) and lemma 2.2.2 (Neapolitan, 2003, p. 86). In the other direction, let *gp*' be the DAG pattern faithful to *P*, the two conditions confirm *gp*'=*gp* according to theorem 2.3.2 (Neapolitan, 2003, p. 87).

Recall that

[

According to theorem 2.3.2 (Neapolitan, 2003, p. 87), two DAGs $G_1$ and $G_2$ are Markov equivalent if and only if they have the same links (edges without regard for direction) and the same set of uncoupled head-to-head meetings.

According to lemma 2.2.2 (Neapolitan, 2003, p. 86), given a DAG *G* = (*V*, *E*) and an uncoupled meeting *X*–*Z*–*Y*, the three following statements are equivalent:

1. *X*–*Z*–*Y* is a head-to-head meeting.
2. There exists a set not containing *Z* that d-separates *X* and *Y*.
3. All sets containing *Z* do not d-separate *X* and *Y*.

]

In general, if faithfulness condition is satisfied, independence $I_G(A, B \mid C)$ and dependence $NI_G(A, B \mid C)$ in DAG *G* are as same as independence $I_P(A, B \mid C)$ and dependence $NI_P(A, B \mid C)$ in joint probability distribution *P*, respectively. More specifically, absence of an edge $I_G(\{X\}, \{Y\})$ and presence of an edge $NI_G(\{X\}, \{Y\})$ in *G* implies direct independence $I_P(\{X\}, \{Y\})$ and direct dependence $NI_P(\{X\}, \{Y\})$ in *P*. Faithfulness condition makes the pair (*G*, *P*) are matched totally, which causes that the (*G*, *P*) is perfect.

## 2.5. Other advanced concepts

Markov condition is essential to BN. Without Markov condition, it is almost impossible to research and apply BN. Faithfulness condition is much stronger than Markov condition, which is essential to structure learning but it costs us dear to obtain faithfulness condition. There is an intermediary condition between Markov condition and faithfulness condition. It is called minimality condition, which is stronger than Markov condition but weaker than faithfulness condition. According to definition 2.5.1 (Neapolitan, 2003, p. 104), given a DAG $G = (V, E)$ and a joint probability distribution $P$, we say $(G, P)$ satisfies **minimality condition** if the two following conditions hold:

1. $(G, P)$ satisfies Markov condition.
2. If any edge is removed from $G$ then, $(G, P)$ is no longer satisfies Markov condition.

**Example 2.5.1.** For illustrating minimality condition, given three DAGs and a joint probability distribution $P(X, Y, Z)$ shown in figure 2.4.2, we will test whether they satisfy minimality condition (Neapolitan, 2003, p. 104).

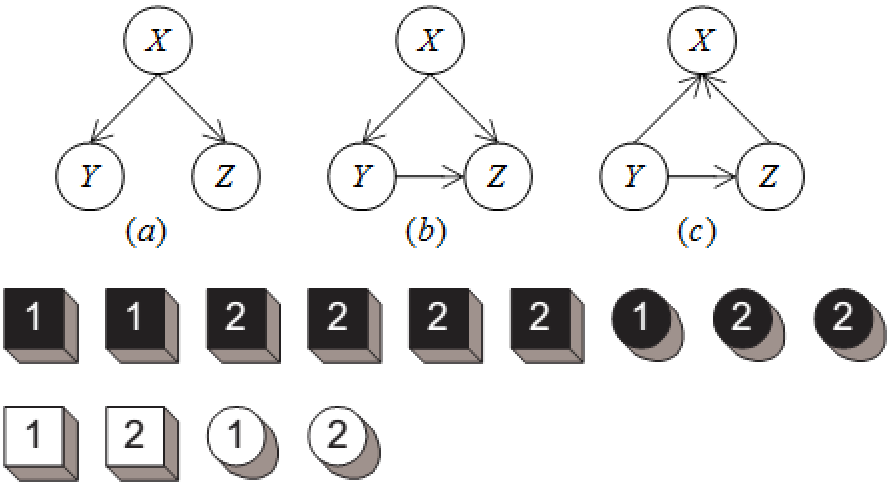


**Figure 2.5.1.** Three DAGs for testing minimality condition

Variable $X$, $Y$, and $Z$ represents colored objects, numbered objects, and square-round objects, respectively (Neapolitan, 2003, p. 11). There are such 13 objects shown in figures 2.2.2 and 2.4.1 (Neapolitan, 2003, p. 12). Values of $X$, $Y$, and $Z$ are defined as seen in table 2.1.1 (Neapolitan, 2003, p. 32):

| | |
|---|---|
| $X$=1 | All black objects |
| $X$=0 | All white objects |
| $Y$=1 | All object named “1” |
| $Y$=0 | All object named “2” |
| $Z$=1 | All square objects |
| $Z$=0 | All round objects |

The joint probability distribution $P(X, Y, Z)$ assigns a probability of 1/13 to each object. In other words, $P(X, Y, Z)$ is determined as relative frequencies among such 13 objects. For example, $P(X=1, Y=1, Z=1)$ is probability of objects which are black, named “1”, and square. There are 2 such objects and hence, $P(X=1, Y=1, Z=1) = 2/13$. From table 2.1.3, $P(Y, Z \mid X)$ equals $P(Y \mid X)P(Z \mid X)$ for all values of $X$, $Y$, and $Z$, which implies $I_P(\{Y\}, \{Z\} \mid \{X\})$ holds. Moreover, it is easy to assert that $I_P(\{Y\}, \{Z\} \mid \{X\})$ is the unique conditional independence from $P(X, Y, Z)$. The DAG in figure 2.5.1 (a) satisfies minimality condition with $P$ because if we remove edges $X \to Y$ and $X \to Z$, we have new d-separations $I_G(\{Y\}, \{X, Z\})$ and $I_G(\{Z\}, \{X, Y\})$ but we do not have conditional independencies $I_P(\{Y\}, \{X, Z\})$ and $I_P(\{Z\}, \{X, Y\})$ hold in $P$. The DAG in figure 2.5.1 (b) does not satisfy the minimality condition with *P* because if remove the edge *Y*→*Z*, the new d-separation $I_G(\{Y\}, \{Z\}|\{X\})$ was also hold in *P* with $I_P(\{Y\}, \{Z\}|\{X\})$. The DAG in figure 2.5.1 (c) does satisfy the minimality condition with *P* because no edge can be removed without creating a d-separation that is not an independency in *P* (Neapolitan, 2003, p. 104)■

That minimality condition is stronger than Markov condition but weaker than faithfulness condition is confirmed by theorem 2.5.1 (Neapolitan, 2003, p. 105). According to this theorem, given a DAG *G* = (*V*, *E*) and a joint probability distribution *P*, if (*G*, *P*) satisfies faithfulness condition, then (*G*, *P*) satisfies minimality condition but the reversed direction is not asserted because there is a case that (*G*, *P*) satisfies minimality condition but does not satisfy faithfulness condition. For example, both DAGs in figures 2.5.1 (a) and 2.5.1 (c) satisfy minimality condition but only the DAG in figure 2.5.1 (a) satisfies faithfulness condition with *P*.

Theorem 2.5.2 (Neapolitan, 2003, p. 105) is applied to create a BN that satisfies minimality condition from a set of nodes and a join probability distribution. According to this theorem, given a set of nodes *V* and a join probability distribution *P*, we create an arbitrary ordering of nodes in *V*. For each *X* in *V*, let $B_X$ be the set of all nodes that come before *X* in the ordering and let $PA_X$ be a minimal subset of $B_X$ such that

$$I_P(\{X\}, B_X|PA_X)$$

We then create a DAG *G* by placing an edge from each node in $PA_X$ to *X*. As a result, (*G*, *P*) satisfies minimality condition. Moreover, if *P* is strictly positive (that is, there is no probability values equal 0), $PA_X$ is unique relative to the ordering. Note, there may be many $PA_X$ which are minimal subsets of $B_X$ such that $I_P(\{X\}, B_X|PA_X)$ if *P* is not strictly positive. It is interesting to recognize that theorem 2.1.2 (Neapolitan, 2003, p. 37) is applied to create a BN that satisfies Markov condition and theorem 2.5.2 (Neapolitan, 2003, p. 105) is applied to create a BN that satisfies minimality condition.

**Example 2.5.2.** Given a BN (*G*, *P*) satisfies faithfulness condition where the DAG *G* is shown in figure 2.5.2 (a) (Neapolitan, 2003, p. 107).

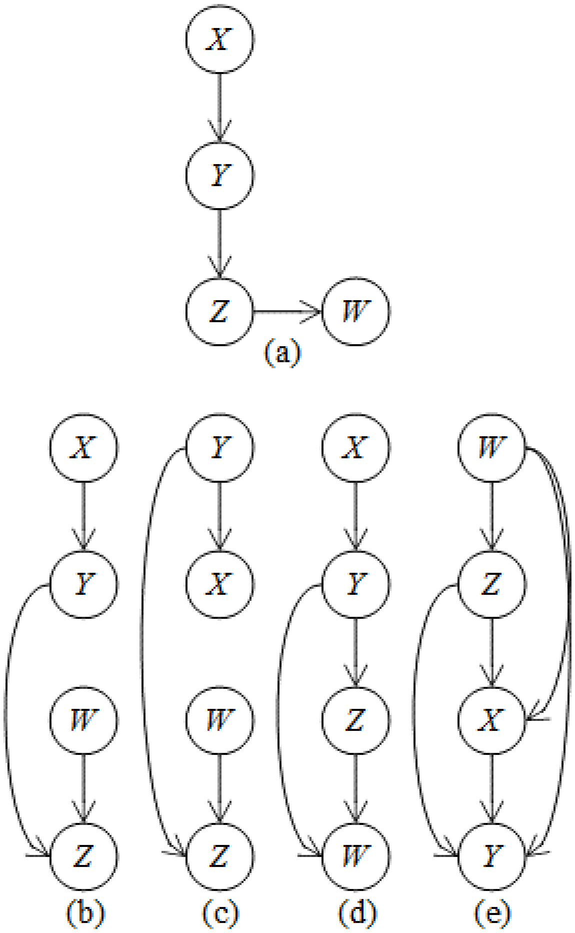


**Figure 2.5.2.** DAGs created to satisfy minimality condition

By applying theorem 2.5.2 (Neapolitan, 2003, p. 105) for four orderings (*X*, *Y*, *W*, *Z*), (*Y*, *X*, *W*, *Z*), (*X*, *Y*, *Z*, *W*), and (*W*, *Z*, *X*, *Y*), we get four DAGs (b), (c), (d), and (e), respectively. Of course, the four DAGs satisfy minimality condition with *P*. If *P* is strictly positive, each of these DAGs is uniquely relative to its ordering (Neapolitan, 2003, p. 107)■

Because a BN can have a lot of nodes, a given node *X* can be affected by far nodes, for instance, calculating probability of *X* needs to instantiate and browse over many other nodes far from *X*. Thus, Markov blanket and Markov boundary are concepts to obtain nodes which are close to *X*, which aims to reduce computation cost related to *X*.

According to definition 2.5.2 (Neapolitan, 2003, p. 108), given a set of nodes *V*, a joint probability distribution *P*, and a node *X* in *V* then, a **Markov blanket** $M_X$ of *X* is any set of nodes such that *X* is conditionally independent from all other variables given $M_X$ as follows:

$$I_P(\{X\}, V\backslash(M_X \cup \{X\})|M_X)$$

Note, the sign "\" denotes the subtraction (excluding) in set theory (Wikipedia, Set (mathematics), 2014).

According to theorem 2.5.3 (Neapolitan, 2003, p. 108), if (*G*, *P*) satisfies the Markov condition, then for each variable *X*, the set which includes all parents of *X*, children of *X*, and parents of children of *X* is a Markov blanket of *X*. Hence, this theorem links Markov condition with Markov blanket.

**Example 2.5.3.** Given a BN ($G$, $P$) satisfies Markov condition where the DAG $G$ is shown in figure 2.5.3 (Neapolitan, 2003, p. 108). According theorem 2.5.3 (Neapolitan, 2003, p. 108), Markov blanket of $X$ is {$T$, $Y$, $Z$}■

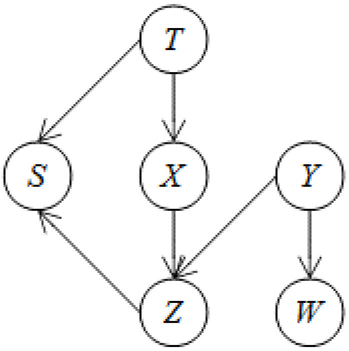


**Figure 2.5.3.** Markov blanket and Markov boundary

According to definition 2.5.3 (Neapolitan, 2003, p. 109), given a set of nodes $V$, a joint probability distribution $P$, and a node $X$ in $V$ then, **Markov boundary** of $X$ is any Markov blanket such that none of its proper subsets is a Markov blanket of $X$. So, Markov boundary can be considered as smallest Markov blanket. According to theorem 2.5.4 (Neapolitan, 2003, p. 109), if ($G$, $P$) satisfies the faithfulness condition, then for each variable $X$, the set which includes all parents of $X$, children of $X$, and parents of children of $X$ is the *unique Markov boundary* of $X$. Hence, theorem 2.5.4 links faithfulness condition with Markov boundary. According to theorem 2.5.5 (Neapolitan, 2003, p. 109), if $P$ is a strictly positive probability distribution of the variables in set $V$, then for each $X$ in $V$ there is a unique Markov boundary of $X$. Note, there is no probability values equal 0 in strictly positive probability distribution.

**Example 2.5.4.** Suppose a BN ($G$, $P$) satisfies faithfulness condition where the DAG $G$ is shown in figure 2.5.3 (Neapolitan, 2003, p. 108). According theorem 2.5.4 (Neapolitan, 2003, p. 109), Markov boundary of $X$ is {$T$, $Y$, $Z$}■

Now advanced concepts relevant to BN were introduced in sections 1 and 2. Three main subjects of BN are inference, parameter learning, and structure learning which are mentioned in successive sections 3, 4, and 5. Recall that this report focuses on discrete BN. In other words, nodes are random discrete variables attached to CPTs. Especially, random binary variables are preferred.

# 3. Inference

The essence of Bayesian reference is to compute the posterior probabilities of nodes given evidences. Equation 1.10 is the base of simple inference, which is an extension of Bayes' rule specified equation 1.1. Note that evidences or conditions are also nodes which are observed and have concrete values. Going back example "wet grass" in section 1, 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 $C$, $R$, $S$, according to equation 1.12 with note that equation 1.12 is an interpretation of equation 1.10.

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

Here we make clear equation 1.0 again. Let $V$ = {$X_1$, $X_2$,…, $X_n$} be a whole set of nodes. Let $D$ = {$X_{m+1}$, $X_{m+2}$,…, $X_{m+u}$} be a set of evidences, $D \subset V$. Let $d$ = {$x_{m+1}$, $x_{m+2}$,…, $x_{m+u}$} be the instantiation of $D$. In general case, the marginal probability of $X_i = x_i$ is:

$$P(X_i = x_i, D = d) = \sum_{V \backslash (\{X_i\} \cup D)} P(X_1, X_2, \dots, x_i, \dots, d, \dots, X_n)$$

Where $P(X_1, X_2,\dots, X_n)$ is the joint probability distribution. The marginal probability of $D = d$ is:

$$P(D = d) = \sum_{X \backslash D} P(X_1, X_2, \dots, d, \dots, X_n)$$

The posterior probability of $X_i = x_i$ given $D = d$ is:

$$P(X_i = x_i | D = d) = \frac{P(X_i = x_i, D = d)}{P(D = d)} = \frac{\sum_{V \backslash (\{X_i\} \cup D)} P(X_1, X_2, \dots, x_i, \dots, d, \dots, X_n)}{\sum_{X \backslash D} P(X_1, X_2, \dots, d, \dots, X_n)} \quad (3.1)$$

The equation 3.1 is the basic idea of simple inference, which is an interpretation of equation 1.10. 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 distribution has many terms, brute force method for determining combinations of such operations is impossible. There are three main approaches that improve this computation:

- Taking advantages of Markov condition: Pearl's message propagation (Pearl, 1986), (Neapolitan, 2003, pp. 126-156) is well-known algorithm.
- Taking advantages of the structure of DAG: Noisy OR-gate model (Neapolitan, 2003, pp. 156-160) and Junction Tree (Neapolitan, 2003, p. 161) are well-known algorithms.
- Reducing the amount of numeric operations computed in marginal probability: Symbolic probabilistic inference (SPI) algorithm (Neapolitan, 2003, pp. 162-170) is the well-known algorithm which finds optimal factoring for marginal probability computation.

## 3.1. Markov condition based inference

When a $(G, P)$ satisfies Markov condition, each node of $G$ is associated with a CPT. The well-known algorithm that takes advantages of conditional independences entailed by Markov condition is Pearl's message propagation algorithm (Pearl, 1986). Pearl's algorithm starts with a $(G, P)$ where the DAG $G$ is a directed tree. Suppose the DAG $G = (E, V)$ is a directed 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 $PA_X$ be children and parents of $X$, respectively. Note, both $C_X$ and $PA_X$ exclude $X$. Let $R$ be root node. Let $O$ be evidence node, $O \in D$. In figure 3.1.1, $N_X$ is green and $D_X$ is red.

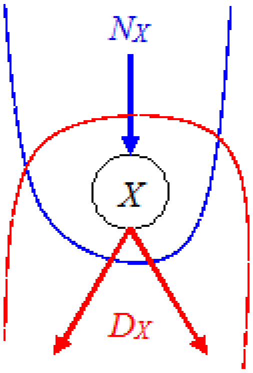


**Figure 3.1.1.** $X$, $D_X$, and $N_X$

The essence of inference is to compute the posterior probability $P(X|D)$ for every $X$. We have (Neapolitan, 2003, p. 128):

$P(X|D) = P(X|D_X, N_X)$

$$= \frac{P(D_X, N_X|X)P(X)}{P(D_X, N_X)}$$

(Due to Bayes' rule)

$$= \frac{P(D_X|X)P(N_X|X)P(X)}{P(D_X, N_X)}$$

(Because $D_X$ and $N_X$ are conditionally independent given $X$)

$$= P(D_X|X)\frac{P(N_X|X)P(X)}{P(N_X)}\frac{P(N_X)}{P(D_X, N_X)}$$

$$= P(D_X|X)P(X|N_X)\frac{P(N_X)}{P(D_X, N_X)}$$

(Due to Bayes' rule)

$$= \alpha P(D_X|X)P(X|N_X) \blacksquare$$

Where $\alpha = \frac{P(N_X)}{P(D_X, N_X)}$ is the constant independent from $X$. Let $\lambda(X) = P(D_X|X)$ and $\pi(X) = P(X|N_X)$, equation 3.1.1 is used to calculate the posterior probability $P(X|D)$, which is the base of Pearl's message propagation algorithm (Neapolitan, 2003, p. 128).

$$P(X|D) = \alpha\lambda(X)\pi(X) \quad (3.1.1)$$

The $\lambda(X)$ and $\pi(X)$ are called $\lambda$ value and $\pi$ value of $X$, respectively. For each child $Y$ of $X$, let $\lambda_Y(X)$ be $\lambda$ message that is propagated up from $Y$ to $X$. Note that $\lambda_Y(X)$ is conditional probability of $D_Y$ given $X$. Equation 3.1.2 specifies the $\lambda$ message $\lambda_Y(X)$.

$$\lambda_Y(X) = P(D_Y|X) = \sum_Y \lambda(Y)P(Y|X) \quad (3.1.2)$$

Following is the proof of equation 3.1.2.

$$\lambda_Y(X) = P(D_Y|X) = \sum_Y P(D_Y|X,Y)P(Y|X) = \sum_Y P(D_Y|Y)P(Y|X) = \sum_Y \lambda(Y)P(Y|X) \blacksquare$$

For each parent $Z$ of $X$, let $\pi_X(Z)$ be $\pi$ message that is propagated down from $Z$ to $X$. Note that $\pi_X(Z)$ is conditional probability of $X$ given $N_X$. Equation 3.1.3 specifies the $\pi$ message $\pi_X(Z)$.

$$\begin{gathered}\pi_X(Z) = \pi(Z)\prod_{K\in C_Z\backslash\{X\}} \lambda_K(Z) \\ \pi_X(Z) \propto P(Z|N_X)\end{gathered} \quad (3.1.3)$$

Where the notation "∝" denote proportion and $C_Z\backslash\{X\}$ is the set of $Z$'s children except $X$. Following is the proof of equation 3.1.3.

$$P(Z|N_X) = P\left(Z\middle|N_Z, \bigcap_{K\in C_Z\backslash\{X\}} D_K\right)$$

$$= \frac{P\left(N_Z, \bigcap_{K\in C_Z\backslash\{X\}} D_K \middle|Z\right)P(Z)}{P\left(N_Z, \bigcap_{K\in C_Z\backslash\{X\}} D_K\right)}$$

(Due to Bayes' rule)

$$= \frac{P(N_Z|Z)P\left(\bigcap_{K\in C_Z\backslash\{X\}} D_K \middle|Z\right)P(Z)}{P\left(N_Z, \bigcap_{K\in C_Z\backslash\{X\}} D_K\right)}$$

(Because $N_Z$ and $\bigcap_{K\in C_Z\backslash\{X\}} D_K$ are conditionally independent give $Z$)

$$= \frac{P(Z|N_Z)P(N_Z)}{P(Z)}\frac{P\left(\bigcap_{K\in C_Z\backslash\{X\}} D_K \middle|Z\right)P(Z)}{P\left(N_Z, \bigcap_{K\in C_Z\backslash\{X\}} D_K\right)}$$

$$= P(Z|N_Z)P\left(\cap_{K\in C_Z\backslash\{X\}} D_K \,|Z\right)\frac{P(N_Z)}{P\left(N_Z, \cap_{K\in C_Z\backslash\{X\}} D_K\right)} = kP(Z|N_Z)P\left(\cap_{K\in C_Z\backslash\{X\}} D_K \,|Z\right)$$

(Where $k = \frac{P(N_Z)}{P\left(N_Z, \cap_{K\in C_Z\backslash\{X\}} D_K\right)}$ is the constant independent from *X* and *Z*)

$$= k\pi(Z)\prod_{K\in C_Z\backslash\{X\}} P(D_K|Z)$$

(Because *Z*'s children are mutually independent)

$$= k\pi(Z)\prod_{K\in C_Z\backslash\{X\}} \lambda_K(Z)$$

$$\propto \pi(Z)\prod_{K\in C_Z\backslash\{X\}} \lambda_K(Z) = \pi_X(Z)\blacksquare$$

Don't worry about $\pi_X(Z)$ which is proportioned to $P(Z|N_X) = k\pi(Z)\prod_{K\in C_Z\backslash\{X\}}\lambda_K(Z)$ and the posterior probability *P*(*X*|*D*) itself is also proportioned to $\lambda(X)$ and $\pi(X)$ via constant $\alpha$. These constants will be eliminated when *P*(*X*|*D*) is normalized. For example, given binary random variable *X*, if $P(X=1 \mid D) = \alpha p_1$ and $P(X=0 \mid D) = \alpha p_2$, they are normalized as follows.

$$P(X = 1|D) = \frac{\alpha p_1}{\alpha p_1 + \alpha p_2} = \frac{p_1}{p_1 + p_2}$$

$$P(X = 1|D) = \frac{\alpha p_2}{\alpha p_1 + \alpha p_2} = \frac{p_2}{p_1 + p_2}$$

Now we have:

- Value $\lambda(X) = P(D_X|X)$.
- Message $\lambda_Y(X)$ is calculated according to equation 3.1.2 for each $Y \in C_X$.
- Value $\pi(X) = P(X|N_X)$.
- Message $\pi_X(Z)$ is calculated according to equation 3.1.3 for each $Z \in PA_X$.

The $\lambda$ and $\pi$ values will be updated according to $\lambda$ and $\pi$ messages, mentioned later. Whenever evidence $O \in D$ occurs, Pearl's algorithm propagates downwards $\pi$ message and propagates upwards $\lambda$ message in order to update $\lambda$ value and $\pi$ 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, as seen in figure 3.1.2.

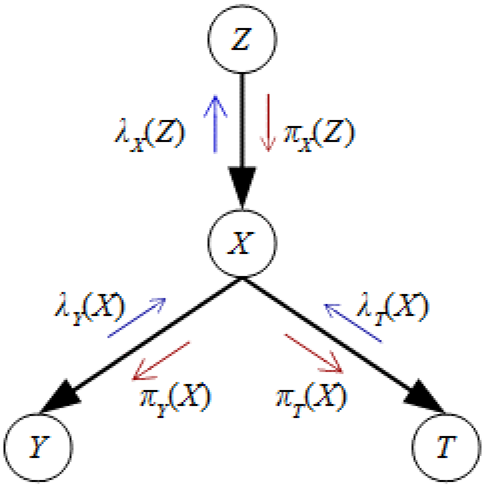


**Figure 3.1.2.** Pearl propagation algorithm (*X* is focused node)

Please pay attention to four following cases when updating $\lambda$ value and $\pi$ value at certain variable $X$ (Neapolitan, 2003, pp. 127-128):

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:
   $\lambda(X) = P(\emptyset|X) = 1$ due to $D_X = \emptyset$.
   $\pi(X)$ is computed as if $X$ were intermediate variable according to case 4.
   $P(X|D) = \alpha\pi(X)$.
3. If $X \notin D$ and $X$ is root then:
   $\lambda(X)$ is computed as if $X$ were intermediate variable according case 4.
   $\pi(X) = P(X|\emptyset) = P(X)$.
   $P(X|D) = \alpha\lambda(X)P(X)$.
4. If $X \notin D$ and $X$ is intermediate variable then, $\lambda(X)$ and $\pi(X)$ are computed according equations 3.1.4 and 3.1.5. Later on $P(X|D)$ is calculated according to equation 3.1.1, $P(X|D)= \alpha\lambda(X)\pi(X)$.

Hence, equation 3.1.4 is used to update and $\lambda$ value based on $\lambda$ message.

$$\lambda(X) = P(D_X|X) = \prod_{Y \in C_X} \lambda_Y(X) \tag{3.1.4}$$

Following is the proof of equation 3.1.4.

$$\lambda(X) = P(D_X|X) = P\left(\cap_{Y \in C_X} D_Y \,\middle|X\right) = \prod_{Y \in C_X} P(D_Y|X) = \prod_{Y \in C_X} \lambda_Y(X) \blacksquare$$

(Because $X$'s children are mutually independent).

Equation 3.1.5 is used to update $\pi$ value according to $\pi$ message.

$$\begin{gathered}\pi(X) = \sum_Z P(X|Z)\pi_X(Z) \\ \pi(X) \propto P(X|N_X)\end{gathered} \tag{3.1.5}$$

Following is the proof of equation 3.1.5.

$$P(X|N_X) = \sum_Z P(X|Z, N_X)P(Z|N_X) = \sum_Z P(X|Z)P(Z|N_X) \propto \sum_Z P(X|Z)\pi_X(Z) = \pi(X) \blacksquare$$

Where $Z$ is parent of $X$. The C-like pseudo-code for Pearl's algorithm shown below includes four functions:

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

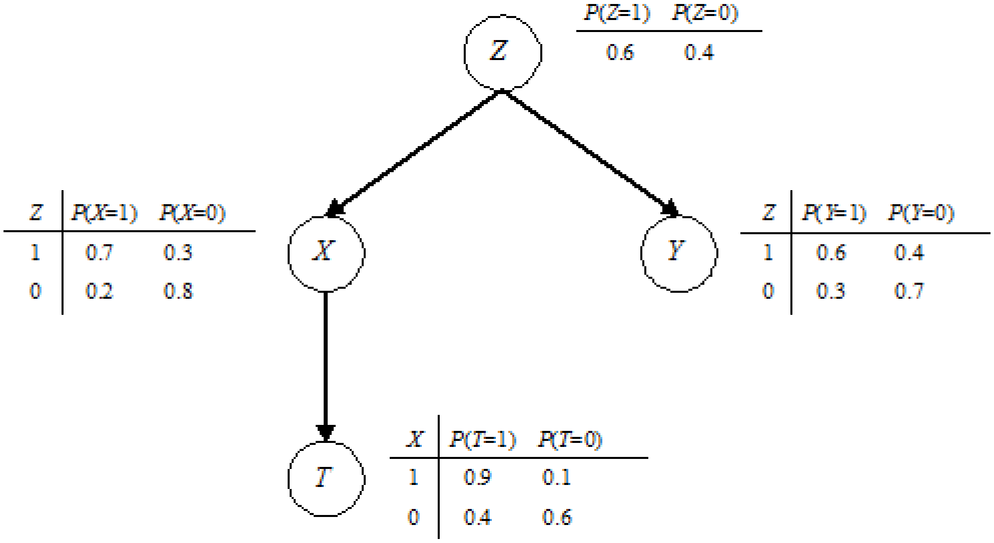
- Function "*propagate_down_π_message*" computes $\pi$ value, posterior probability of current node, and continues to propagate downwards $\pi$ message by calling itself. Process of propagation stops when there is no node to be propagated.

Followings are descriptions of these functions.

```
void init($G$, $D$)
{
  $D$=Ø;
  for each $X \in V$
  {
    $\lambda(X) = 1$;                    //due to $D$ = Ø
    for each parent $Z$ of $X$     //propagate up $\lambda$ message
      $\lambda_X(Z) = 1$;                //due to $D$ = Ø
  }
  $P(R|D) = P(R)$;                 //posterior probability of root node
  $\pi(R) = P(R)$;                  //$\pi$ value

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

void update($O$, $o$)
{
  $D = D \cup O$;
  $\lambda(O=o) = \pi(O=o) = P(O=o|D) = 1$;      //due to $O \in D$ and $O=o$
  $\lambda(O \neq o) = \pi(O \neq o) = P(O \neq o|D) = 0$;      //due to $O \in D$ and $O \neq o$

  if $O \neq R$ and $O$'s parent $Z \notin D$            //$O$ 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 $K \notin D$ //browse $O$'s children
    propagate_down_π_message($O$, $K$);
}

void propagate_up_λ_message($Y$, $X$)
{
  $\lambda_Y(X) = \sum_Y \lambda(Y)P(Y|X)$;  //$Y$ propagate upwards $\lambda$ message
  $\lambda(X) = \prod_{Y \in C_X} \lambda_Y(X)$;       //update $\lambda$ value
  $P(X|D) = \alpha\lambda(X)\pi(X)$;          //compute posterior probability of $X$
  normalize $P(X|D)$;             //eliminate constant $\alpha$

  if $X \neq R$ and $X$'s parent $Z \notin D$
    propagate_up_ λ_message($X$, $Z$);

  for each child $K$ of $X$ such that $K \neq Y$ and $K \notin D$  //browse $O$'s children
    propagate_down_π_message($X$, $K$);
}
```

```
void propagate_down_π_message(Z, X)
{
  $\pi_X(Z) = \pi(Z)\prod_{K \in C_Z \setminus \{X\}} \lambda_K(Z)$;  //Y propagate downwards π message
  $\pi(X) = \sum_Z P(X|Z)\pi_X(Z)$;           //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_down_π_message(X, K);
}
```

**Example 3.1.1.** Given a (*G*, *P*) shown in figure 3.1.3 where DAG *G* is a directed tree satisfying Markov condition and each binary node has a CPT, suppose evidence *X* has value 1. Hence, we need to compute posterior probabilities of *T*, *Y*, *Z* in condition *X*=1.



| P(Z=1) | P(Z=0) |
|---|---|
| 0.6 | 0.4 |

| Z | P(X=1) | P(X=0) |
|---|---|---|
| 1 | 0.7 | 0.3 |
| 0 | 0.2 | 0.8 |

| Z | P(Y=1) | P(Y=0) |
|---|---|---|
| 1 | 0.6 | 0.4 |
| 0 | 0.3 | 0.7 |

| X | P(T=1) | P(T=0) |
|---|---|---|
| 1 | 0.9 | 0.1 |
| 0 | 0.4 | 0.6 |

**Figure 3.1.3.** Bayesian network with CPTs

Firstly, function "*init*" is called to initialize network.

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

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

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

Calling *propagate_down_π_message*(*Z*, *X*)

Calling *propagate_down_π_message*(*Z*, *Y*)

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

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

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

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

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

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

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

Normalizing $P(X)$

$P(X=1) = (\alpha 0.5) / (\alpha 0.5 + \alpha 0.5) = 0.5$

$P(X=0) = (\alpha 0.5) / (\alpha 0.5 + \alpha 0.5) = 0.5$

Calling *propagate_down_π_message*(*X*, *T*)

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

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

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

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

$\pi(T=0) = P(T=0|X=1)\pi_T(X=1) + P(T=0|X=0)\pi_T(X=0) = 0.1*0.5 + 0.6*0.5 = 0.40$

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

$P(T=0) = \alpha\lambda(T=0)\pi(T=0) = \alpha*1*0.40 = \alpha 0.40$

Normalizing $P(T)$

$P(T=1) = (\alpha 0.65) / (\alpha 0.65 + \alpha 0.40) = 0.62$

$P(T=0) = (\alpha 0.40) / (\alpha 0.65 + \alpha 0.40) = 0.38$

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

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

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

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

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

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

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

Normalizing $P(Y)$

$P(Y=1) = (\alpha 0.45) / (\alpha 0.45 + \alpha 0.68) = 0.4$

$P(Y=0) = (\alpha 0.68) / (\alpha 0.45 + \alpha 0.68) = 0.6$

The initialized Bayesian network is shown in figure 3.1.4.

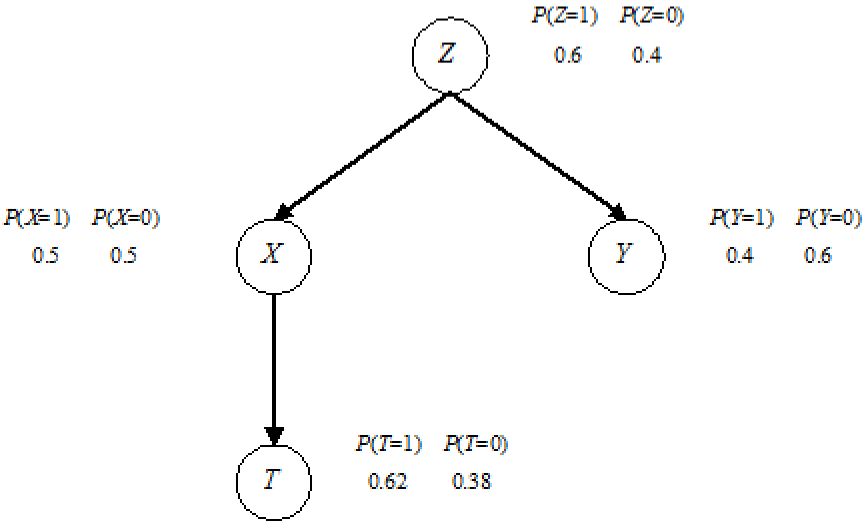


**Figure 3.1.4.** Initialized Bayesian network

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

$D = D \cup \{X\} = \emptyset \cup \{X\} = \{X\}$

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:

$\lambda_X(Z=1) = \lambda(X=1)P(X=1|Z=1) + \lambda(X=0)P(X=0|Z=1) = 1*0.7 + 0*0.3 = 0.7$

$\lambda(Z=1) = \lambda_X(Z=1)\lambda_Y(Z=1) = 0.7*1 = 0.7$

$P(Z=1|d) = \alpha\lambda(Z=1)\pi(Z=1) = \alpha 0.7*0.6 = \alpha 0.42$

$\lambda_X(Z=0) = \lambda(X=1)P(X=1|Z=0) + \lambda(X=0)P(X=0|Z=0) = 1*0.2 + 0*0.8 = 0.2$

$\lambda(Z=0) = \lambda_X(Z=0)\lambda_Y(Z=0) = 0.2*1 = 0.2$

$P(Z=0|d) = \alpha\lambda(Z=0)\pi(Z=0) = \alpha 0.2*0.4 = \alpha 0.08$

Normalizing $P(Z)$

$P(Z=1|d) = (\alpha 0.42) / (\alpha 0.42 + \alpha 0.08) = 0.84$

$P(Z=0|d) = (\alpha 0.08) / (\alpha 0.42 + \alpha 0.08) = 0.16$

Calling *propagate_down_π_message* ($Z$, $Y$)

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

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

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

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

$\pi(Y=0) = P(Y=0|Z=1)\pi_X(Z=1) + P(Y=0|Z=0)\pi_X(Z=0) = 0.3*0.6 + 0.8*0.4 = 0.50$

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

$P(Y=0) = \alpha\lambda(Y=0)\pi(Y=0) = \alpha*1*0.5 = \alpha 0.50$

Normalizing $P(Y)$
$P(Y=1) = (\alpha 0.48) / (\alpha 0.48 + \alpha 0.50) = 0.49$
$P(Y=0) = (\alpha 0.50) / (\alpha 0.48 + \alpha 0.50) = 0.51$

Then function *propagate_down_π_message*($X$, $T$) is executed
$\pi_T(X=1) = \pi(X=1) = 1$
$\pi_T(X=0) = \pi(X=0) = 0$

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

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

Normalizing $P(T)$
$P(T=1) = (\alpha 0.9) / (\alpha 0.9 + \alpha 0.1) = 0.9$
$P(T=0) = (\alpha 0.1) / (\alpha 0.9 + \alpha 0.1) = 0.1$

Finally, all posterior probabilities are computed as in figure 3.1.5■

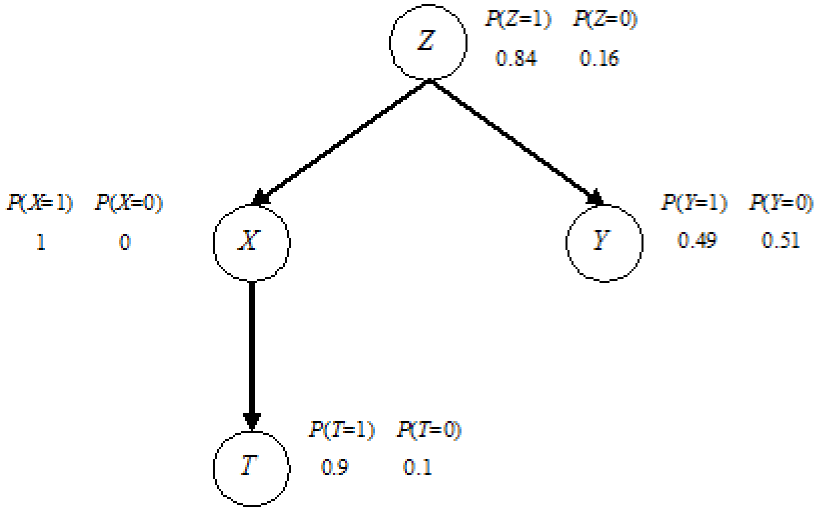


**Figure 3.1.5.** All posterior probabilities are computed after running Pearl algorithm (*X* is evidence)

## 3.2. DAG based inference

DAG based inference algorithms take advantages of straightforward structure of DAG without cycle. This approach starts with as simplest DAG which has many parents (input nodes) and one child (output node) following the *noisy OR-gate* model in which the output value becomes *true* (1) if there is at least one of inputs being *true* (1). Figure 3.2.1 is an example of noisy OR-gate network with cause-effect relationships.

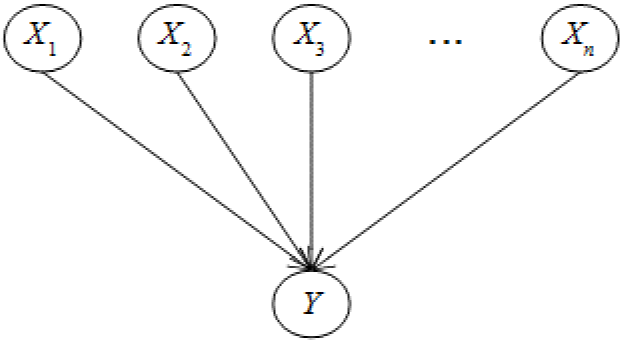


**Figure 3.2.1.** Noisy OR-gate network with cause-effect relationships

Noisy OR-gate model can be used to build a simple BN when there are too many parent nodes or each parent node has too many discrete values. For example, if there are *n* binary parent nodes, given binary child node need a CPT having $2^n$ entries. In this case, noisy OR-gate algorithm establishes one equation to determine the CPT of child node, which is an interesting result.

Suppose every node is binary, noisy OR-gate inference in Bayesian network simulates electronic circuit based on three assumptions:

- *Cause inhibition*: Given a cause-effect relationship denoted by edge *X*→*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* (exception independence): Inhibitions are mutually independent. For example, inhibition $I_1$ of $X_1$ is independent from inhibition $I_2$ of $X_2$.
- *Noisy OR-gate condition* (accountability): Suppose we have a set of cause-effect relationships in which *Y* is the effect of many causes $X_1$, $X_2$,…, $X_n$ (see figure 3.2.1). 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*:
$$\exists i: X_i = 0 \vee I_1 = 1 \Rightarrow Y = 0$$

Suppose we have *n* causes $X_1$, $X_2$,…, $X_n$ 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 accountability 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 "noisy 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*.

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

Figure 3.2.2 shows the noisy OR-gate model of the network shown in figure 3.2.1

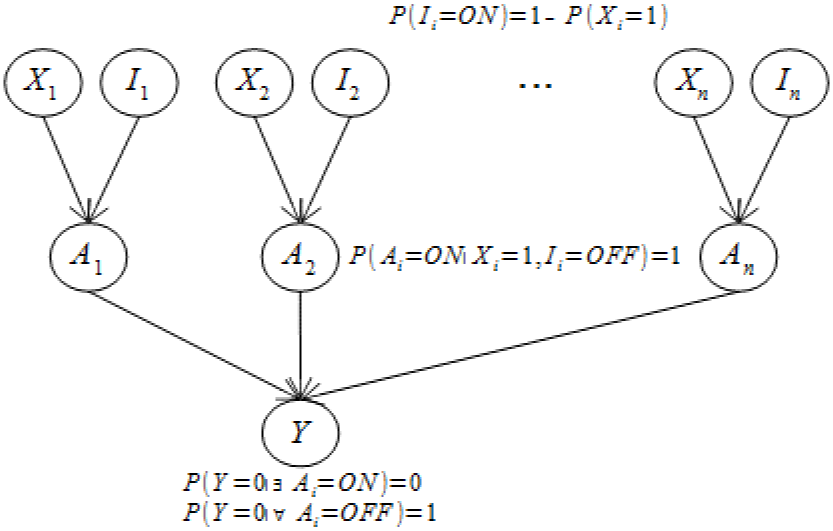

$P(I_i=ON)=1-P(X_i=1)$
$X_1$
$I_1$
$X_2$
$I_2$
…
$X_n$
$I_n$
$A_1$
$A_2$
$P(A_i=ON|X_i=1,I_i=OFF)=1$
$A_n$
$Y$
$P(Y=0|\exists\ A_i=ON)=0$
$P(Y=0|\forall\ A_i=OFF)=1$

**Figure 3.2.2.** 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 of $P(X_i=1)$ according to equation 3.2.1.

$$P(I_i=ON) = 1 - P(X_i=1) \qquad (3.2.1)$$

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 $P(Y| X_1, X_2,\ldots, X_n)$. We have:

$$P(Y = 0|X_1, X_2, \ldots, X_n)$$

$$= \sum_{A_1,A_2,\ldots,A_n} P(Y = 0|A_1, A_2, \ldots, A_n)P(A_1, A_2, \ldots, A_n|X_1, X_2, \ldots, X_n)$$

(Due to total probability rule)

$$= \sum_{A_1,A_2,\ldots,A_n} \left(P(Y = 0|A_1, A_2, \ldots, A_n)\prod_{i=1}^{n} P(A_i|X_1, X_2, \ldots, X_n)\right)$$

(Because $A_i$ (s) are mutually independent)

$$= \sum_{A_1,A_2,\ldots,A_n} \left(P(Y = 0|A_1, A_2, \ldots, A_n)\prod_{i=1}^{n} P(A_i|X_i)\right)$$

(Because each $A_i$ is only dependent on $X_i$)

$$= \prod_{i=1}^{n} P(A_i = OFF|X_i)$$

(Due to $P(Y=0|\exists\ A_i=ON) = 0$ and $P(Y=0|\forall\ A_i=OFF) = 1$)

$$= \prod_{i=1}^{n} \big(P(A_i = OFF|X_i, I_i = ON)P(I_i = ON) + P(A_i = OFF|X_i, I_i = OFF)P(I_i = OFF)\big)$$

$$
\begin{aligned}
&= \prod_{i \in O}^{n} \big( P(A_i = OFF | X_i = 1, I_i = ON) P(I_i = ON) \\
&\qquad\qquad + P(A_i = OFF | X_i = 1, I_i = OFF) P(I_i = OFF) \big) \\
&+ \prod_{i \notin O}^{n} \big( P(A_i = OFF | X_i = 0, I_i = ON) P(I_i = ON) \\
&\qquad\qquad + P(A_i = OFF | X_i = 0, I_i = OFF) P(I_i = OFF) \big) \\
&= \prod_{i \in O}^{n} \big( 1 * P(I_i = ON) + 0 * P(I_i = OFF) \big) + \prod_{i \notin O}^{n} \big( 1 * P(I_i = ON) + 1 * P(I_i = OFF) \big) \\
&= \prod_{i \in O}^{n} \big( 1 - P(X_i = 1) \big) + \prod_{i \notin O}^{n} \big( 1 - P(X_i = 1) + P(X_i = 1) \big) \\
&= \prod_{i \in O}^{n} \big( 1 - P(X_i = 1) \big) \blacksquare
\end{aligned}
$$

In general, we have equation 3.2.2 to specify noisy OR-gate inference.

$$
\begin{gathered}
P(Y = 0 | X_1, X_2, \dots, X_n) = \prod_{i \in O} \big( 1 - P(X_i) \big) \\
P(Y = 1 | X_1, X_2, \dots, X_n) = 1 - \prod_{i \in O} \big( 1 - P(X_i) \big)
\end{gathered}
\tag{3.2.2}
$$

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

**Example 3.2.1.** Given cause-effect relationship shown in figure 3.2.3. Given prior probabilities of causes $X_1$=1, $X_2$=1, $X_3$=1 are 0.2, 0.5, 0.3, respectively. For example, we need to compute the conditional probability of effect $P(Y=1|X_1=1, X_2=0, X_3=1)$.

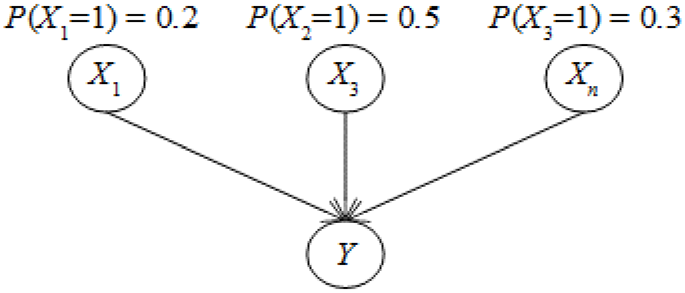


**Figure 3.2.3.** Noisy OR-gate inference example.

Applying equation 3.2.2, we have $P(Y=1| X_1=1, X_2=0, X_3=1) = 1 - (1 - P(X_1=1))(1 - P(X_3=1)) = 1 - 0.8*0.7 = 0.44$■

## 3.3. Optimal factoring based inference

Given a ($G$, $P$) (Neapolitan, 2003, p. 162) where $G$ is the DAG shown in figure 3.3.1 and $P$ is the joint probability distribution $P(X, Y, Z, W, T) = P(T | Z)P(W | Y, Z)P(Y | X)P(Z | X)P(X)$. Note, all nodes are binary variables.

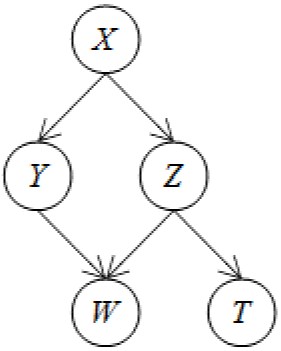


**Figure 3.3.1.** A DAG used for illustrating optimal factoring based inference

Suppose $W$ becomes evidence and we need to make an inference on $T$ which is to compute the posterior probability $P(T \mid W)$ according to equation 1.10 and 3.1 as follows (Neapolitan, 2003, p. 162):

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

We survey the numerator of the equation above as an example of optimal factoring based inference.

$$P(T,W)=\sum_{X,Y,Z}P(T|Z)P(W|Y,Z)P(Y|X)P(Z|X)P(X) \qquad (3.3.1)$$

Because the sum is over 3 binary variables ($X$, $Y$, $Z$) and there are 4 multiplications in $P(T, W)$, it requires $2^3$ * 4 = 32 multiplications to calculate one $P(T, W)$. Because $T$ and $W$ has 4 possible values, it requires totally 32*4 = 128 multiplications to calculate all values of $P(T, W)$. The computation cost will be save if each product is not re-calculated when it is needed. For example, we factorize $P(T, W)$ into 4 products as follows (Neapolitan, 2003, p. 163):

$$P(T,W)=\sum_{X,Y,Z}\left[\left[\left[[P(T|Z)P(W|Y,Z)]P(Y|X)\right]P(Z|X)\right]P(X)\right]$$

For illustration, suppose we create 4 buckets for such 4 products. Of course, such buckets are pseudo.

$$P(T,W)=\sum_{X,Y,Z}\underbrace{\left[\underbrace{\left[\underbrace{\left[\underbrace{[P(T|Z)P(W|Y,Z)]}_{bucket1}P(Y|X)\right]}_{bucket2}P(Z|X)\right]}_{bucket3}P(X)\right]}_{bucket4}$$

So, we have *bucket*1 = $\{P(T \mid Z)P(W \mid Y, Z)\}$ for the first product, *bucket*2 = $\{bucket1*P(Y|X)\}$ for the second product, *bucket*3 = $\{bucket2*P(Z|X)\}$ for the third product, and *bucket*4 = $\{bucket3*P(X)\}$ for the fourth product. After these products are calculated, they are stored in buckets. *Bucket*4 contains all possible values of $P(T, W)$. Now we determine how many multiplications used for these buckets. The *bucket*1 as the first product $P(T \mid Z)P(W \mid Y, Z)$ requires $2^4$ = 16 multiplications (combinations) because it involves 4 binary variables. The *bucket*2 as the second product *bucket*1*$P(Y|X)$ = $P(T \mid Z)P(W \mid Y, Z)P(Y \mid X)$ requires $2^5$ = 32 multiplications (combinations) because it involves 5 binary variables. The *bucket*3 as the third product *bucket*2*$P(Z|X)$ = $P(T \mid Z)P(W \mid Y, Z)P(Y \mid X)P(Z \mid X)$ requires $2^5$ = 32 multiplications (combinations) because it involves 5 binary variables. The *bucket*4 as the fourth product *bucket*3*$P(X)$ = $P(T \mid Z)P(W \mid Y, Z)P(Y \mid X)P(Z \mid X)P(X)$ requires $2^5$ = 32 multiplications (combinations) because it involves 5 binary variables. In general, *P*(*T*, *W*) requires |*bucket*1| + |*bucket*2| + |*bucket*3| + |*bucket*4| = 16 + 32 + 32 + 32 = 112 multiplications. We save 16 multiplications when *P*(*T*, *W*) needs 128 multiplications as usual. Note, additions in sigma sums are not concerned.

We can save more multiplications by summing over a variable when such variable no longer appears in remaining terms as follows (Neapolitan, 2003, p. 163):

$$P(T,W)=\sum_X \underbrace{\left[P(X)\sum_Z \underbrace{\left[P(Z|X)\sum_Y \underbrace{\left[\underbrace{[P(T|Z)P(W|Y,Z)]}_{bucket1}P(Y|X)\right]}_{bucket2}\right]}_{bucket3}\right]}_{bucket4}$$

The *bucket*1 requires $2^4$ = 16 multiplications because it involves 4 binary variables. The *bucket*2 requires $2^5$ = 32 multiplications because it involves 5 binary variables. The *bucket*3 requires $2^4$ = 16 multiplications because it only involves 4 binary variables when we sum *Y* out before taking *bucket*3. The *bucket*4 require $2^3$ = 8 multiplications because it only involves 3 binary variables when we sum *Z* out before taking *bucket*4. In general, *P*(*T*, *W*) requires |*bucket*1| + |*bucket*2| + |*bucket*3| + |*bucket*4| = 16 + 32 + 16 + 8 = 72 multiplications.

The other factorization of *P*(*T*, *W*) is optimal as follows:

$$P(T,W)=\sum_Z \underbrace{\left[P(T|Z)\sum_Y \underbrace{\left[P(W|Y,Z)\sum_X \underbrace{\left[P(Y|X)\underbrace{[P(Z|X)P(X)]}_{bucket1}\right]}_{bucket2}\right]}_{bucket3}\right]}_{bucket4} \qquad (3.3.2)$$

Now the *bucket*1 requires $2^2$ = 4 multiplications because it involves 2 binary variables. The *bucket*2 requires $2^3$ = 8 multiplications because it involves 3 binary variables. The *bucket*3 requires $2^3$ = 8 multiplications because it only involves 3 binary variables when we sum *X* out before taking *bucket*3. The *bucket*4 require $2^3$ = 8 multiplications because it only involves 3 binary variables when we sum *Y* out before taking *bucket*4. In general, *P*(*T*, *W*) requires |*bucket*1| + |*bucket*2| + |*bucket*3| + |*bucket*4| = 4 + 8 + 8 + 8 = 28 multiplications. Such the number of multiplications is now minimum for the aforementioned *P*(*W*, *T*). In general, we need to find out a way to factorize the product *P*(*T* | *W*) into a minimum number of multiplications as equation 3.3.2. This is the *Optimal Factoring Problem* given by Shachter, D’Ambrosio, and Del Favero (Shachter, D'Ambrosio, & Del Favero, 1990).

According to definition 3.3.1 (Neapolitan, 2003, p. 163), a *factoring instance F* = {*V*, *S*, *Q*} is defined as a triple consisting of:

1. A set of *n* variables *V*= {$X_1$, $X_2$,…, $X_n$}
2. A set of *m* sub-sets *S* = {$S_{\{1\}}$, $S_{\{2\}}$,…, $S_{\{m\}}$} where $S_{\{i\}} \subseteq V$
3. A *target set* $Q \subseteq V$

According to definition 3.3.2 (Neapolitan, 2003, p. 164), the factoring *α* of *S* is a binary tree satisfying three following properties (Neapolitan, 2003, p. 164):

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

Given *F*, the cost of factoring *α* denoted $\mu_\alpha(F)$ is three following steps (Neapolitan, 2003, p. 164):

1. All non-leave nodes are determined according to equation 3.3.3.

$$S_{I\cup J}=\left(S_I\cup S_J\right)\backslash W_{I\cup J}\ where\ W_{I\cup J}=\left\{w:\left(\forall k\notin I\cup J,w\notin S_{\{k\}}\right)\ and\ \left(w\notin Q\right)\right\} \quad (3.3.3)$$

   Note, the sign "\" denotes the subtraction (excluding) in set theory (Wikipedia, Set (mathematics), 2014).

2. The cost of each node is computed according to equations 3.3.4.

$$\begin{gathered}\mu_\alpha\left(S_{\{j\}}\right)=0\\ \mu_\alpha\left(S_{I\cup J}\right)=\mu_\alpha\left(S_I\right)+\mu_\alpha\left(S_J\right)+2^{\left|S_I\cup S_J\right|}\end{gathered} \quad (3.3.4)$$

   Where |.| denotes the cardinality of the set.

3. The cost of factoring $\alpha$ is $\mu_\alpha(F) = \mu_\alpha(S_{\{1,\ldots,\,m\}}))$.

The less the cost $\mu_\alpha(F)$ is, the better factoring $\alpha$ is. Hence, the optimal factoring problem is to find the optimal factoring $\alpha$ for the factoring instance $F$ such that $\mu_\alpha(F)$ is minimal.

When applying optimal factoring problem into Bayesian inference, the set of variables $V$ in $F$ corresponds with nodes in DAG, $S$ corresponds with operands of the marginal probability, and the factoring $\alpha$ corresponds with the factorization of such probability. The cost of factoring instance $\mu_\alpha(F)$ is equal to the number of multiplications. The problem becomes easy when we find out the best tree $\alpha$ having least $\mu_\alpha(F)$ and compute the marginal probability with the same ordering of multiplications to this tree.

**Example 3.3.1.** According to definition 3.3.1 (Neapolitan, 2003, p. 163), let the following factoring instance model the marginal probability $P(T, W)$ specified by equation 3.3.1 for the DAG shown in figure 3.3.1 as follows (Neapolitan, 2003, p. 164):

- Let $n = 5$ and $V = \{X, Y, Z, W, T\}$.
- Let $m = 5$ and $S_{\{1\}} = \{X\}$, $S_{\{2\}} = \{X, Z\}$, $S_{\{3\}} = \{X, Y\}$, $S_{\{4\}} = \{Y, Z, W\}$, and $S_{\{5\}} = \{Z, T\}$.
- Let $Q = \{W, T\}$.

It is easy to recognize that $S_{\{1\}}$, $S_{\{2\}}$, $S_{\{3\}}$, $S_{\{4\}}$, and $S_{\{5\}}$ correspond with $P(X)$, $P(Z \mid X)$, $P(Y \mid X)$, $P(W \mid Y, \mathrm{Z})$, and $P(T \mid Z)$, respectively. Suppose the optimal factorizing $\alpha$ shown in figure 3.3.2 (Neapolitan, 2003, p. 165) corresponds with the factorization of the marginal probability $P(W, T)$ shown in equation 3.3.2 with note that Shachter, D'Ambrosio, and Del Favero (Shachter, D'Ambrosio, & Del Favero, 1990) proposed a linear time algorithm to find out such $\alpha$.

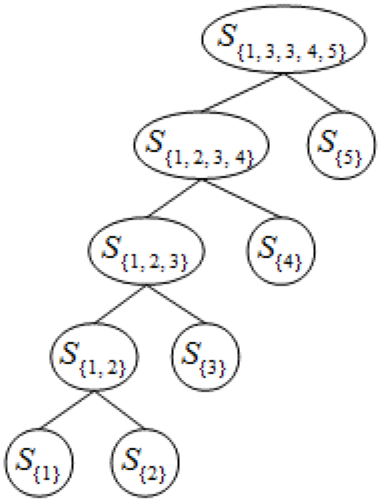


**Figure 3.3.2.** An optimal factorizing

We will know the cost $\mu_\alpha(F)$ of the factorizing $\alpha$ shown in figure 3.3.2 is 28 as aforementioned. In fact, we have (Neapolitan, 2003, p. 166):

$$S_{\{1,2\}}=S_{\{1\}}\cup S_{\{2\}}\backslash W_{\{1,2\}}=\{X\}\cup\{X,Z\}\backslash\emptyset=\{X,Z\}$$

$$S_{\{1,2,3\}} = S_{\{1,2\}} \cup S_{\{3\}} \backslash W_{\{1,2,3\}} = \{X,Z\} \cup \{X,Y\} \backslash \{X\} = \{Y,Z\}$$
$$S_{\{1,2,3,4\}} = S_{\{1,2,3\}} \cup S_{\{4\}} \backslash W_{\{1,2,3,4\}} = \{Y,Z\} \cup \{Y,Z,W\} \backslash \{X,Y\} = \{Z,W\}$$
$$S_{\{1,2,3,4,5\}} = S_{\{1,2,3,4\}} \cup S_{\{5\}} \backslash W_{\{1,2,3,4,5\}} = \{Z,W\} \cup \{Z,T\} \backslash \{X,Y,Z\} = \{W,T\}$$

The costs are computed as follows:

$$\mu_\alpha\left(S_{\{1,2\}}\right) = \mu_\alpha\left(S_{\{1\}}\right) + \mu_\alpha\left(S_{\{2\}}\right) + 2^2 = 0 + 0 + 4 = 4$$
$$\mu_\alpha\left(S_{\{1,2,3\}}\right) = \mu_\alpha\left(S_{\{1,2\}}\right) + \mu_\alpha\left(S_{\{3\}}\right) + 2^3 = 4 + 0 + 8 = 12$$
$$\mu_\alpha\left(S_{\{1,2,3,4\}}\right) = \mu_\alpha\left(S_{\{1,2,3\}}\right) + \mu_\alpha\left(S_{\{4\}}\right) + 2^3 = 12 + 0 + 8 = 20$$
$$\mu_\alpha\left(S_{\{1,2,3,4,5\}}\right) = \mu_\alpha\left(S_{\{1,2,3,4\}}\right) + \mu_\alpha\left(S_{\{5\}}\right) + 2^3 = 20 + 0 + 8 = 28$$

So, the cost of the factoring $\alpha$ is $\mu_\alpha(F) = \mu_\alpha(S_{\{1, 2, 3, 4, 5\}})) = 28$■

Shortly, after giving the optimal factoring problem, Shachter, D'Ambrosio, and Del Favero (Shachter, D'Ambrosio, & Del Favero, 1990) proposed a linear time algorithm which solves the optimal factoring problem when the DAG is singly-connected. Because their algorithm combines both the symbolic reasoning and the numeric computation for doing probabilistic inference, it is called *Symbolic Probabilistic Inference* (*SPI*) algorithm.

# 4. Parameter learning

We turn back Bayesian inference introduced in equation 1.1 here. As a convention, uppercase letters such as *X*, *Y*, and *Z* often denote random variable whereas lowercase letters such as *x*, *y*, and *z* often denote instances or values of random variables. According to Bayesian approach, parameters such as mean $\mu$ and variance $\sigma^2$ of normal distribution and probability *p* of binomial distribution are random variables too. These random variables are commonly denoted Θ, which are hypotheses according to equation 1.1. Prior distribution (prior probability) is denoted $P(\Theta \mid \xi)$ where $\xi$ denotes background knowledge about Θ. Note that $\xi$ is often parameter of the prior distribution and so it can be called hyper-parameter of prior distribution. For example, if Θ follows beta distribution, its prior distribution is:

$$P(\Theta|\xi) = \mathrm{beta}(\Theta|a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\Theta^{a-1}(1-\Theta)^{b-1}$$

Where $\xi = (a, b)^T$ are two parameters of such prior (beta) distribution. Note that Γ(.) is gamma function:

$$\Gamma(x) = \int_0^{+\infty} t^{x-1}e^{-t}\mathrm{d}t$$

For another example, if Θ is mean $\mu$ and it follows normal distribution, its prior distribution is:

$$P(\Theta|\xi) = \mathcal{N}(\Theta|\mu_0,\sigma_0^2) = \frac{1}{\sqrt{2\pi\sigma_0^2}}e^{-\frac{(\Theta-\mu_0)^2}{2\sigma_0^2}}$$

Where $\xi = (\mu_0, \sigma_0^2)^T$ are mean and variance of such prior (normal) distribution. Given sample $D = \{X_1, X_2,\ldots, X_m)$ consisting *m* observations (evidences) $X_i$ with note that *m* can be 1. Let *X* be theoretical random variable that represents all $X_i$. Equation 4.1 specifies posterior distribution (posterior probability) of Θ, according to Bayes' rule (Heckerman, 1995, p. 6) (Wikipedia, Bayesian inference, 2006).

$$P(\Theta|D,\xi) = \frac{P(D|\Theta)P(\Theta|\xi)}{P(D)} \quad (4.1)$$

Where $P(D \mid \Theta)$ is likelihood function of Θ and *P*(*D*) is marginal probability of sample. If all observations $X_i$ are independent and identically distributed (iid) random variables, we have:

$$P(D|\Theta)=\prod_{X_i}P(X_i|\Theta)$$

The probability $P(X_i \mid \Theta)$ is likelihood function of Θ in simplest case of Bayesian inference specified by equation 1.1. Of course, $P(X_i \mid \Theta)$ can follow a distribution which is different from prior distribution $P(\Theta \mid \xi)$.

If posterior distribution $P(\Theta \mid D, \xi)$ has the same form of prior distribution $P(\Theta \mid \xi)$, such posterior distribution and prior distribution are called *conjugate distributions* (conjugate probabilities) and $P(\Theta \mid \xi)$ is called *conjugate prior* (Wikipedia, Conjugate prior, 2018) for likelihood function $P(D \mid \Theta)$. For example, if prior distribution $P(\Theta \mid \xi)$ is beta distribution and likelihood function $P(D \mid \Theta)$ follows binomial distribution then, posterior distribution $P(\Theta \mid D, \xi)$ is beta distribution too and hence, $P(\Theta \mid \xi)$ and $P(\Theta \mid D, \xi)$ are conjugate distributions. Shortly, whether posterior distribution and prior distribution are conjugate distributions depends on prior distribution and likelihood function.

Equation 4.1 is an extension of equation 1.1. Note, equation 4.1 is written fully as follows:

$$P(\Theta|D,\xi)=\frac{P(D|\Theta,\xi)P(\Theta|\xi)}{P(D|\xi)}$$

However $P(D \mid \Theta, \xi) = P(D \mid \Theta)$ and $P(D \mid \xi) = P(D)$ because $D$ is only dependent on Θ. Marginal probability $P(D)$ is expectation of likelihood function $P(D \mid \Theta)$ given prior probability $P(\Theta \mid \xi)$.

$$P(D)=\sum_{\Theta}P(D|\Theta)P(\Theta|\xi)\text{ if }\Theta\text{ is discrete}$$

$$P(D)=\int_{\Theta}P(D|\Theta)P(\Theta|\xi)\mathrm{d}\Theta\text{ if }\Theta\text{ is continuous}$$

Anyway equation 4.2 specifies marginal probability $P(D)$.

$$P(D)=E(P(D|\Theta)|\Theta)\tag{4.2}$$

Equation 4.1 which defines posterior probability of parameter Θ is used to assess hypothesis Θ after surveying sample $D$. This is a so-called *Bayesian inference*.

Suppose there is a requirement of predicting possibility of a new observation $X_{m+1}$ given previous sample $D$ with note that $X_{m+1}$ is independent from $D$. In other words, we need to calculate probability $P(X_{m+1} \mid D)$ called updated probability. Following equation specifies *updated probability* or *predictive probability* (Heckerman, 1995, p. 6) (Wikipedia, Bayesian inference, 2006).

$$P(X_{m+1}|D)=\int_{\Theta}P(X_{m+1}|D,\Theta)P(\Theta|D,\xi)\mathrm{d}\Theta=\int_{\Theta}P(X_{m+1}|\Theta)P(\Theta|D,\xi)\mathrm{d}\Theta$$

Because we consider $X_{m+1}$ and $X$ are formal variables, we have $P(X_{m+1} \mid D) = P(X \mid D)$ and $P(X_{m+1} \mid \Theta) = P(X \mid \Theta)$. Therefore, equation 4.3 specifies updated probability in general.

$$P(X|D)=\int_{\Theta}P(X|\Theta)P(\Theta|D,\xi)\mathrm{d}\Theta\tag{4.3}$$

According to equation 4.3, updated probability $P(X \mid D)$ is expectation of probability $P(X \mid \Theta)$ given posterior distribution $P(\Theta \mid D)$. Equation 4.3 establishes a so-called *Bayesian updating* or *Bayesian prediction*. Probability $P(X \mid \Theta)$ is always determined because it is probability of observation.

Given a sample $D = \{X_1, X_2,\ldots, X_m)$ where all observations $X_i$ are independent and identically distributed (iid) random variables, there is a requirement of estimating parameter (hypothesis) Θ according to Bayesian inference. Let $\widehat{\Theta}$ denote a Bayesian estimate of Θ. How to calculate the estimate $\widehat{\Theta}$ is called *Bayesian estimation* or *Bayesian learning* which is main subject of this section

"Parameter learning". There are some methods to determine $\widehat{\Theta}$. The most popular method is Maximum A Posteriori (MAP) estimation (Wikipedia, Maximum a posteriori estimation, 2017). According to MAP, $\widehat{\Theta}$ is a maximizer of posterior distribution given sample *D*.

$$\widehat{\Theta} = \underset{\Theta}{\operatorname{argmax}} P(\Theta|D,\xi) = \underset{\Theta}{\operatorname{argmax}} \frac{P(D|\Theta)P(\Theta|\xi)}{P(D)}$$

Because *P*(*D*) is constant regarding Θ, we have:

$$\widehat{\Theta} = \underset{\Theta}{\operatorname{argmax}} P(D|\Theta)P(\Theta|\xi)$$

For convenience, we take natural logarithm of *P*(*D*|Θ)*P*(Θ|*ξ*), which produce equation 4.4 to determine $\widehat{\Theta}$ according to MAP.

$$\begin{gathered}\widehat{\Theta} = \underset{\Theta}{\operatorname{argmax}} P(D|\Theta)P(\Theta|\xi) = \underset{\Theta}{\operatorname{argmax}}\big(l(\Theta|\xi)\big) \\ l(\Theta|\xi) = \log\big(P(D|\Theta)P(\Theta|\xi)\big) = \log\big(P(D|\Theta)\big) + \log\big(P(\Theta|\xi)\big)\end{gathered} \tag{4.4}$$

As a convention, the function *l*(Θ|*ξ*) is called posterior log-likelihood function. Hence, MAP is extension of maximum likelihood estimation (MLE) method. If equation 4.4 is complicated, some approximate methods such as Newton-Raphson, gradient descent, and Lagrange duality to solve equation 4.4. By the simplest way, $\widehat{\Theta}$ is solution of the equation formed by setting the (partial) first-order of posterior log-likelihood function *l*(Θ|*ξ*) to be zero as follows:

$$\frac{\partial l(\Theta|\xi)}{\partial\Theta} = 0$$

Other estimation methods use so-called loss functions. Squared-error loss function is defined according to equation 4.5 (Walpole, Myers, Myers, & Ye, 2012, p. 717):

$$l_{se}\big(\Theta,\widehat{\Theta}\big) = \big(\Theta - \widehat{\Theta}\big)^2 \tag{4.5}$$

The mean of posterior distribution *P*(Θ | *D*, *ξ*) is a Bayesian estimate of Θ under squared-error loss function, according to equation 4.6 (Walpole, Myers, Myers, & Ye, 2012, p. 717). In other words, such mean minimizes squared-error loss function.

$$\widehat{\Theta} = \underset{\Theta}{\operatorname{argmin}}\, l_{se}\big(\Theta,\widehat{\Theta}\big) = \int_{\Theta} \Theta P(\Theta|D,\xi)\mathrm{d}\Theta = E(\Theta) \tag{4.6}$$

Absolute loss function is defined according to equation 4.7 (Walpole, Myers, Myers, & Ye, 2012, p. 718):

$$l_a\big(\Theta,\widehat{\Theta}\big) = \big|\Theta - \widehat{\Theta}\big| \tag{4.7}$$

Median of posterior distribution *P*(Θ | *D*, *ξ*) is a Bayesian estimate of Θ under absolute loss function, according to equation 4.8 (Walpole, Myers, Myers, & Ye, 2012, p. 718). In other words, such median minimizes absolute loss function.

$$\begin{gathered}\widehat{\Theta} = \underset{\Theta}{\operatorname{argmin}}\, l_a\big(\Theta,\widehat{\Theta}\big) \\ P\big(\Theta \le \widehat{\Theta}\big|D,\xi\big) = P\big(\Theta \ge \widehat{\Theta}\big|D,\xi\big) = \frac{1}{2}\end{gathered} \tag{4.8}$$

Therefore, equations 4.4, 4.6 and 4.8 are popular equations for *Bayesian parameter estimation*. If posterior distribution *P*(Θ | *D*, *ξ*) is symmetric, equations 4.6 and 4.8 produces the same estimate $\widehat{\Theta}$.

Now we survey a common case of Bayesian inference in which *D* is **binomial sample**. At that time every $X_i$ is binary random variable and likelihood function *P*(*D* | Θ) is specified by equation 4.9 (Heckerman, 1995, p. 6):

$$\begin{aligned}&P(D|\Theta) = \Theta^s(1-\Theta)^t\\&P(D) = E(P(D|\Theta)|\Theta) = E(\Theta^s(1-\Theta)^t)\\&\Theta = P(X=1|\Theta)\\&P(X=1) = E(\Theta)\end{aligned} \quad (4.9)$$

Where *s* and *t* are the numbers of $X_i = 1$ and $X_i = 0$, respectively. The *s* and *t* are sufficient statistics of binomial sampling. Note, *X* is theoretical random variable that represents all $X_i$. Parameter Θ = *P*(*X* = 1 | Θ) is probability of *X* = 1 and it has prior probability *P*(Θ|*ξ*). Therefore,

$$P(X=1) = \int_\Theta P(X=1|\Theta)P(\Theta|\xi)\mathrm{d}\Theta = \int_\Theta \Theta P(\Theta|\xi)\mathrm{d}\Theta = E(\Theta)$$

Equation 4.10 (Heckerman, 1995, p. 6), which is a special case of equation 4.1, specifies Bayesian inference (posterior probability of Θ) in case of binomial sampling.

$$P(\Theta|D,\xi) = \frac{\Theta^s(1-\Theta)^t P(\Theta|\xi)}{P(D)} \quad (4.10)$$

Derived from equation 4.3, the updated probability *P*(*X* = 1 | *D*) in case of binomial sampling becomes (Heckerman, 1995, p. 6):

$$P(X=1|D) = \int_\Theta P(X=1|\Theta)P(\Theta|D,\xi)\mathrm{d}\Theta$$

Due to:

$$P(X=1|\Theta) = \Theta$$

Equation 4.11, which is a variant of equation 4.3, specifies updated probability *P*(*X* = 1 | *D*) in case of binomial sampling (Heckerman, 1995, p. 6).

$$P(X=1|D) = \int_\Theta \Theta P(\Theta|D,\xi)\mathrm{d}\Theta = E(\Theta|D) \quad (4.11)$$

Where *E*(Θ | *D*, *ξ*) denotes expectation of Θ given posterior probability *P*(Θ | *D*, *ξ*) specified by equation 4.10. Comparing equation 4.11 and equation 4.6, we recognize that updated probability is the estimate of Θ under squared-error loss function in case of binomial sampling.

$$\widehat{\Theta} = P(X=1|D) = E(\Theta|D)$$

Suppose Θ is distributed according to beta distribution, its prior probability is specified by equation 4.12.

$$\begin{aligned}P(\Theta|\xi) &= \mathrm{beta}(\Theta|a,b) = \mathrm{beta}(\Theta;\mathrm{a},\mathrm{b}) = \beta(\Theta|a,b) = \beta(\Theta;a,b)\\&= \frac{\Gamma(N)}{\Gamma(a)\Gamma(b)}\Theta^{a-1}(1-\Theta)^{b-1}\end{aligned} \quad (4.12)$$

Where,

$$a > 0, b > 0, N = a + b \text{ and } 0 \le \Theta \le 1$$

So $\xi = (a, b)^T$ contains two parameters of such prior beta distribution. Note, we use two notations "beta" and "*β*" to denote beta distribution. Note, Γ(.) denotes gamma function (Neapolitan, 2003, p. 298) which is essentially an integral approximated to factorial function according to equation 4.13.

$$\Gamma(x) = \int_0^{+\infty} t^{x-1}e^{-t}\mathrm{d}t \quad (4.13)$$

It is conventional that $e^{(.)}$ and *exp*(.) denote exponent function and *e* ≈ 2.71828 is Euler's number. If *x* is positive integer, gamma function in equation 4.13 is equivalent to factorial function,

$$\Gamma(x) = (x-1)!$$

There is an important property of gamma function which is expressed as follows (Neapolitan, 2003, p. 298):

$$\frac{\Gamma(x+1)}{\Gamma(x)} = x$$

Figure 4.1 shows beta density function with various parameters *a* and *b*. Beta functions *β*(*x*;2,2), *β*(*x*;4,2), and *β*(*x*;2,4) are drawn as black line, green line, and red line, respectively.

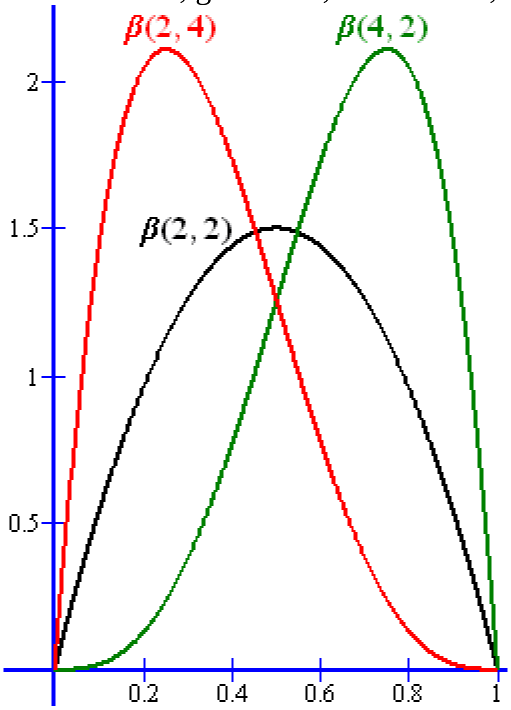


**Figure 4.1.** Beta density functions with various parameters *a* and *b*

In beta density function, there are "*a*" successful outcomes (for example, *x* =1) in "*a*+*b*" trials. The higher value of "*a*" is, the higher ratio of success is, so, the graph leans forward right. The higher value of "*a*+*b*" is, the more the mass concentrates around *a*/(*a*+*b*) and the narrower the graph is.

Theoretical mean and variance of beta distribution is specified by equation 4.14.

$$E(\Theta) = \int_0^1 \Theta\beta(\Theta; a, b)\mathrm{d}\Theta = \frac{a}{N}$$
$$Var(\Theta) = \int_0^1 \left(\Theta - E(\Theta)\right)^2 \beta(\Theta; a, b)\mathrm{d}\Theta = \frac{ab}{N^2(N+1)} \quad (4.14)$$

Where *N* = *a* + *b*. Of course, we have probability *P*(*X*=1) as follows:

$$P(X=1) = E(\Theta) = \frac{a}{N}$$

Marginal probability *P*(*D*) is calculated as follows:

$$P(D) = \int_0^1 P(D|\Theta)P(\Theta|\xi)\mathrm{d}\Theta = \int_0^1 \Theta^s(1-\Theta)^t\frac{\Gamma(N)}{\Gamma(a)\Gamma(b)}\Theta^{a-1}(1-\Theta)^{b-1}\mathrm{d}\Theta = E(\Theta^s(1-\Theta)^t)$$

$$= \int_0^1 \frac{\Gamma(N)}{\Gamma(a)\Gamma(b)}\Theta^{a+s-1}(1-\Theta)^{b+t-1}\mathrm{d}\Theta$$

$$= \frac{\Gamma(N)}{\Gamma(a)\Gamma(b)}\frac{\Gamma(a+s)\Gamma(b+t)}{\Gamma(N+M)}\int_0^1 \beta(\Theta; a+s, b+t)\mathrm{d}\Theta$$

$$= \frac{\Gamma(N)}{\Gamma(a)\Gamma(b)}\frac{\Gamma(a+s)\Gamma(b+t)}{\Gamma(N+M)} \blacksquare$$

Where $M = s + t$. Shortly, equation 4.15 specifies marginal probability $P(D)$.

$$P(D) = E(\Theta^s(1-\Theta)^t) = \frac{\Gamma(N)}{\Gamma(N+M)}\frac{\Gamma(a+s)\Gamma(b+t)}{\Gamma(a)\Gamma(b)} \tag{4.15}$$

Posterior probability of Θ is re-calculated as follows:

$$P(\Theta|D,\xi) = \frac{\Theta^s(1-\Theta)^tP(\Theta|\xi)}{P(D)} = \frac{\Gamma(N+M)}{\Gamma(N)} * \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+s)\Gamma(b+t)} * \Theta^s(1-\Theta)^tP(\Theta|\xi)$$

$$= \frac{\Gamma(N+M)}{\Gamma(N)} * \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+s)\Gamma(b+t)} * \Theta^s(1-\Theta)^t * \frac{\Gamma(N)}{\Gamma(a)\Gamma(b)} * \Theta^{a-1}(1-\Theta)^{b-1}$$

$$= \frac{\Gamma(N+M)}{\Gamma(a+s)\Gamma(b+t)}\Theta^{a+s-1}(1-\Theta)^{b+t-1} = \beta(\Theta; a+s, b+t) \blacksquare$$

Therefore, in case of binomial sampling, if prior probability of Θ conforms beta distribution beta(Θ | *a*, *b*) then, posterior probability of Θ conforms beta distribution beta(Θ | *a* + *s*, *b* + *t*), which is a beautiful result according to equation 4.16 (Heckerman, 1995, p. 7).

$$P(\Theta|D,\xi) = \beta(\Theta; a+s, b+t) \tag{4.16}$$

Equation 4.16 is a special case of equation 4.10 in case of beta distribution. Because both $P(\Theta \mid \xi)$ and $P(\Theta \mid D, \xi)$ conform beta distribution, they are conjugate probabilities.

Equation 4.17 (Heckerman, 1995, p. 7) specifies updated probability $P(X = 1 \mid D)$ in case of binomial sampling and prior beta distribution.

$$P(X=1|D) = E(\Theta|D) = \int_0^1 \Theta\beta(\Theta; a+s, b+t)\mathrm{d}\Theta = \frac{a+s}{N+M} \tag{4.17}$$

Essentially, equation 4.17 is a special case of equation 4.6 in case of binomial sampling and beta prior distribution, which is used to estimate Θ under squared-error loss function as follows:

$$\widehat{\Theta} = P(X=1|D) = E(\Theta|D) = \frac{a+s}{N+M}$$

Now we survey another case of Bayesian inference in which *D* is **multinomial sample**. At that time every $X_i$ is multinomial random variable which has *r* possible states 1, 2,…, *r*. So, *X* is multinomial random variable (discrete random variable). Parameter Θ is now *r*-dimension vector $\Theta = (\theta_1, \theta_1,\ldots, \theta_r)^T$. Partial parameter $\theta_k$ is specified in equation 4.18 (Heckerman, 1995, p. 9):

$$\begin{gathered}\theta_k = P(X=k|\theta_k) \text{ with } k = 1,2,\ldots,r\\ \sum_{k=1}^{r}\theta_k = 1\\ 0 \le \theta_k \le 1\end{gathered} \tag{4.18}$$

Likelihood function $P(D \mid \Theta)$ is specified by equation 4.19.

$$\begin{aligned}
&P(D|\Theta) = \prod_{k=1}^{r} (\theta_k)^{s_k} \\
&P(D) = E(P(D|\Theta)|\Theta) = E\left(\prod_{k=1}^{r} (\theta_k)^{s_k}\right) \\
&P(X = k) = E(\theta_k)
\end{aligned} \tag{4.19}$$

Where $s_k$ is the number of $X_i = k$. Hence, each $s_k$ is sufficient statistics of multinomial sampling. Recall that $X$ is theoretical random variable that represents all $X_i$. Partial parameter $\theta_k = P(X = k \mid \theta_k)$ is the probability of $X = k$ and it has partial prior probability $P(\theta_k|\xi_k)$ which in turn is calculated based on prior probability $P(\Theta \mid \xi) = P(\theta_1, \theta_1,\ldots, \theta_r \mid \xi_1, \xi_2,\ldots, \xi_r)$. Note, $\xi = (\xi_1, \xi_2,\ldots, \xi_r)^T$ is vector parameter of prior probability $P(\Theta \mid \xi)$ with $\Theta = (\theta_1, \theta_1,\ldots, \theta_r)^T$ and each $\theta_k$ only depends on $\xi_k$.

$$P(\theta_k|\xi_k) = P(\theta_k|\xi) = \int\limits_{\theta_1} \cdots \int\limits_{\theta_{k-1}} \int\limits_{\theta_{k+1}} \cdots \int\limits_{\theta_r} P(\Theta|\xi)\mathrm{d}\theta_1 \ldots \mathrm{d}\theta_{k-1}\mathrm{d}\theta_{k+1} \ldots \mathrm{d}\theta_r$$

Therefore,

$$P(X = k) = \int\limits_{\theta_k} P(X = k|\theta_k)P(\theta_k|\xi_k)\mathrm{d}\theta_k = \int_0^1 \theta_k P(\theta_k|\xi_k)\mathrm{d}\theta_k = E(\theta_k)$$

Equation 4.20 (Neapolitan, 2003, p. 385), which is a special case of equation 4.1, specifies Bayesian inference (posterior probability of $\Theta$) in case of multinomial sampling.

$$P(\Theta|D,\xi) = \frac{P(D|\Theta)P(\Theta|\xi)}{P(D)} = \frac{\left(\prod_{k=1}^{r}(\theta_k)^{s_k}\right)P(\Theta|\xi)}{E\left(\prod_{k=1}^{r}(\theta_k)^{s_k}\right)} \tag{4.20}$$

Derived from equation 4.3, updated probability $P(X = k \mid D)$ in case of multinomial sampling becomes:

$$P(X = k|D) = \int_0^1 P(X = k|\theta_k)P(\theta_k|D,\xi_k)\mathrm{d}\theta_k$$

Due to:

$$P(X = k|\theta_k) = \theta_k$$

Equation 4.21, which is a variant of equation 4.3, specifies updated probability $P(X = k \mid D)$ in case of multinomial sampling.

$$P(X = k|D) = \int_0^1 \theta_k P(\theta_k|D,\xi_k)\mathrm{d}\theta_k = E(\theta_k|D) \tag{4.21}$$

Where $E(\theta_k \mid D, \xi_k)$ denotes expectation of $\theta_k$ given partial posterior probability $P(\theta_k \mid D, \xi_k)$ which in turn is calculated based on posterior probability $P(\Theta \mid D, \xi)$ specified by equation 4.20. Equation 4.21 is extension of equation 4.11, which is used to estimate $\theta_k$ under squared-error loss function in case of multinomial sampling.

The most important problem of multinomial sampling is how to determine prior probability of (vector) parameter $\Theta = (\theta_1, \theta_1,\ldots, \theta_r)^T$. Suppose $\Theta$ is distributed according to *Dirichlet distribution*, its prior probability is specified by equation 4.22.

$$P(\Theta|\xi) = \mathrm{Dir}(\Theta|a_1, a_2, \ldots, a_r) = \frac{\Gamma(N)}{\prod_{k=1}^{r} a_k} \prod_{k=1}^{r} (\theta_k)^{a_k - 1} \tag{4.22}$$

Where,

$$N = \sum_{k=1}^{r} a_k$$
$$a_k > 0$$

Equation 4.22 also specifies *Dirichlet density function* because each $\theta_k$ is continuous. It is easy to recognize that Dirichlet distribution is general case of beta distribution (equation 4.12) in case of multinomial sample (discrete sample). So $\xi = (\xi_1, \xi_2, \ldots, \xi_r)^T = (a_1, a_2, \ldots, a_r)^T$ contains $r$ parameters of such prior Dirichlet distribution. Note, Γ(.) denotes gamma function specified by equation 4.13. Partial prior Dirichlet distribution $P(\theta_k|\xi_k) = P(\theta_k|a_k)$ is marginal probability of Dir(Θ | $a_1, a_2, \ldots, a_r$) over all $a_i$ except $a_k$, which is also beta distribution, specified by equation 4.23.

$$\begin{aligned}P(\theta_k|a_k) &= \int_{\theta_1} \cdots \int_{\theta_{k-1}} \int_{\theta_{k+1}} \cdots \int_{\theta_r} \mathrm{Dir}(\Theta|a_1, a_2, \ldots, a_r)\mathrm{d}\theta_1 \ldots \mathrm{d}\theta_{k-1}\mathrm{d}\theta_{k+1} \ldots \mathrm{d}\theta_r \\ &= \mathrm{beta}(\theta_k|a_k, N - a_k) = \frac{\Gamma(N)}{\Gamma(a_k)\Gamma(N-a_k)}(a_k)^{a_k-1}(1-a_k)^{N-a_k-1}\end{aligned} \tag{4.23}$$

Theoretical means and variances of Dirichlet distribution is specified by equation 4.24.

$$\begin{aligned}E(\theta_k) &= \frac{a_k}{N} \\ Var(\theta_k) &= \frac{a_k(1-a_k)}{N^2(N+1)}\end{aligned} \tag{4.24}$$

Equation 4.24 is extension of equation 4.14. Of course, we have the probability $P(X=k)$ as follows:

$$P(X = k) = E(\theta_k) = \frac{a_k}{N}$$

Marginal probability $P(D)$ with regard to Dirichlet distribution is specified by equation 4.25.

$$P(D) = E\left(\prod_{k=1}^{r}(\theta_k)^{s_k}\right) = \frac{\Gamma(N)}{\Gamma(N+M)}\prod_{k=1}^{r}\frac{\Gamma(a_k+s_k)}{\Gamma(a_k)} \tag{4.25}$$

Where,

$$M = \sum_{k=1}^{r} s_k$$

Recall that $s_k$ is the number of $X_i = k$. Equation 4.25 is extension of equation 4.15. It is easy to determine posterior probability $P(\Theta \mid D, \xi)$ given prior probability $P(\Theta \mid \xi)$ = Dir(Θ | $a_1, a_2, \ldots, a_r$) and marginal probability $P(D)$. As a result, in case of multinomial sampling, posterior probability of Θ conforms Dirichlet distribution Dir(Θ | $a_1+s_1, a_2+s_2, \ldots, a_r+s_r$), according to equation 4.26.

$$P(\Theta|D,\xi) = \mathrm{Dir}(\Theta|a_1+s_1, a_2+s_2, \ldots, a_r+s_r) = \frac{\Gamma(N+M)}{\prod_{k=1}^{r}(a_k+s_k)}\prod_{k=1}^{r}(\theta_k)^{(a_k+s_k)-1} \tag{4.26}$$

Equation 4.26 is general case of equation 4.16 and special case of equation 4.20. Because both $P(\Theta \mid \xi)$ and $P(\Theta \mid D, \xi)$ conform Dirichlet distribution, they are conjugate probabilities.

Equation 4.27 (Heckerman, 1995, p. 9) specifies updated probability $P(X = k \mid D)$ in case of multinomial sampling and prior Dirichlet distribution.

$$P(X = k|D) = E(\theta_k|D) = \frac{a_k+s_k}{N+M} \tag{4.27}$$

Equation 4.27 is special case of equation 4.21. Essentially, equation 4.27 is general case of equation 4.17, which is used to estimate Θ under squared-error loss function.

$$\hat{\theta}_k = P(X = k|D) = E(\theta_k|D) = \frac{a_k + s_k}{N + M} \text{ with } k = 1,2,\dots,r$$

We surveyed Bayesian learning with discrete (binomial and multinomial) sample in which parameter Θ follows beta distribution or Dirichlet distribution. As a beautiful result, the estimate $\widehat{\Theta}$ under squared-error loss function is expectation of Θ, which is also updated probability according to equations 4.6, 4.11, 4.17, and 4.27.

$$\widehat{\Theta} = \underset{\Theta}{\operatorname{argmin}}\, l_{se}\big(\Theta, \widehat{\Theta}\big) = \int_{\Theta} \Theta P(\Theta|D,\xi)\mathrm{d}\Theta = E(\Theta)$$

Now suppose $D = \{X_1, X_2,\dots, X_m)$ is **normal sample** where all observations $X_i$ are independent and identically distributed (iid) random variables following normal distribution with theoretical mean $\mu$ and theoretical variance $\sigma^2$. As usual, let $X$ be theoretical random variable that represents all $X_i$. Note, $X$ is real. Suppose the theoretical variance $\sigma^2$ is not random variable but the theoretical mean $\mu$ is random variable $\Theta = \mu$. Probabilistic density function of $X$ is:

$$P(X) = P(X|\mu,\sigma^2) = \mathcal{N}(X|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(X-\mu)^2}{2\sigma^2}} \quad (4.28)$$

Suppose parameter $\mu$ also conforms normal distribution with theoretical mean $\mu_0$ and theoretical variance $\sigma_0^2$. Prior density function (prior distribution) of $\mu = \Theta$ is:

$$P(\mu|\xi) = \mathcal{N}(\mu|\mu_0,\sigma_0^2) = \frac{1}{\sqrt{2\pi\sigma_0^2}} e^{-\frac{(\mu-\mu_0)^2}{2\sigma_0^2}} \quad (4.29)$$

Of course, we have $\xi = (\mu_0, \sigma_0^2)^T$. It is proved that $P(\mu \mid \xi)$ is conjugate prior if $X$ distributes normally. Although the variance $\sigma^2$ is not random variable but it will be estimated in most suitable way. Let $\hat{\mu}$ and $\hat{\sigma}^2$ be estimates of $\mu$ and $\sigma^2$, respectively. The estimates $\hat{\mu}$ and $\hat{\sigma}^2$ will be calculated according to MAP method with equation 4.4.

$$(\hat{\mu}, \hat{\sigma}^2)^T = \underset{\mu,\sigma^2}{\operatorname{argmax}}\, P(D|\mu,\sigma^2)P(\mu|\xi) = \underset{\mu,\sigma^2}{\operatorname{argmax}}\big(l(\mu,\sigma^2|\xi)\big)$$

We have:

$$P(D|\mu,\sigma^2)P(\mu|\xi) = \left(\prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(X_i-\mu)^2}{2\sigma^2}}\right) * \left(\frac{1}{\sqrt{2\pi\sigma_0^2}} e^{-\frac{(\mu-\mu_0)^2}{2\sigma_0^2}}\right)$$

$$= \frac{1}{\left(\sqrt{2\pi\sigma^2}\right)^n \sqrt{2\pi\sigma_0^2}} \exp\left(-\frac{\sum_{i=1}^m (X_i^2 - 2X_i\mu + \mu^2)}{2\sigma^2} - \frac{\mu^2 - 2\mu\mu_0 + \mu_0^2}{2\sigma_0^2}\right)$$

Let notation "∝" denote proportion. Note, only $\mu$ is random variable. By making proportion, Zhu proved that (Zhu, 2018, pp. 3-4):

$$P(D|\mu,\sigma^2)P(\mu|\xi) \propto \exp\left(-\frac{\sum_{i=1}^m (X_i^2 - 2X_i\mu + \mu^2)}{2\sigma^2} - \frac{\mu^2 - 2\mu\mu_0 + \mu_0^2}{2\sigma_0^2}\right)$$

(Because only $\mu$ is considered as random variable)

$$= \exp\left(-\frac{\sum_{i=1}^m X_i^2 - 2\mu\sum_{i=1}^m X_i + m\mu^2}{2\sigma^2} - \frac{\mu^2 - 2\mu\mu_0 + \mu_0^2}{2\sigma_0^2}\right)$$

$$= \exp\left(-\frac{\sigma_0^2\sum_{i=1}^m X_i^2 - 2\mu\sigma_0^2\sum_{i=1}^m X_i + m\mu^2\sigma_0^2 + \mu^2\sigma^2 - 2\mu\mu_0\sigma^2 - \mu_0^2\sigma^2}{2\sigma^2\sigma_0^2}\right)$$

$$= \exp\left(-\frac{(\sigma^2 + m\sigma_0^2)\mu^2 - 2\mu(\mu_0\sigma^2 + \sigma_0^2\sum_{i=1}^m X_i) + \mu_0^2\sigma^2 + \sigma_0^2\sum_{i=1}^m X_i^2}{2\sigma^2\sigma_0^2}\right)$$

$$= \exp\left(-\frac{\mu^2 - 2\mu\frac{(\mu_0\sigma^2 + \sigma_0^2\sum_{i=1}^m X_i)}{\sigma^2 + m\sigma_0^2} + \frac{\mu_0^2\sigma^2 + \sigma_0^2\sum_{i=1}^m X_i^2}{\sigma^2 + m\sigma_0^2}}{\frac{2\sigma^2\sigma_0^2}{\sigma^2 + m\sigma_0^2}}\right)$$

$$\propto \exp\left(-\frac{\mu^2 - 2\mu\frac{\mu_0\sigma^2 + \sigma_0^2\sum_{i=1}^m X_i}{\sigma^2 + m\sigma_0^2} + \left(\frac{\mu_0\sigma^2 + \sigma_0^2\sum_{i=1}^m X_i}{\sigma^2 + m\sigma_0^2}\right)^2}{\frac{2\sigma^2\sigma_0^2}{\sigma^2 + m\sigma_0^2}}\right)$$

(Because only $\mu$ is considered as random variable)

Let

$$\bar{X} = \frac{1}{m}\sum_{i=1}^{m} X_i$$

We have:

$$P(D|\mu,\sigma^2)P(\mu|\xi) \propto \exp\left(-\frac{\mu^2 - 2\mu\frac{\mu_0\sigma^2 + m\bar{X}\sigma_0^2}{\sigma^2 + m\sigma_0^2} + \left(\frac{\mu_0\sigma^2 + m\bar{X}\sigma_0^2}{\sigma^2 + m\sigma_0^2}\right)^2}{\frac{2\sigma^2\sigma_0^2}{\sigma^2 + m\sigma_0^2}}\right)$$

$$= \exp\left(-\frac{\left(\mu - \frac{\sigma^2\mu_0 + m\sigma_0^2\bar{X}}{\sigma^2 + m\sigma_0^2}\right)^2}{\frac{2\sigma^2\sigma_0^2}{\sigma^2 + m\sigma_0^2}}\right)$$

Let

$$\begin{cases} \hat{\mu} = \dfrac{\sigma^2\mu_0 + m\sigma_0^2\bar{X}}{\sigma^2 + m\sigma_0^2} \\ \hat{\sigma}^2 = \dfrac{\sigma^2\sigma_0^2}{\sigma^2 + m\sigma_0^2} \end{cases}$$

We have:

$$P(D|\mu,\sigma^2)P(\mu|\xi) \propto \exp\left(-\frac{(\mu - \hat{\mu})^2}{2\hat{\sigma}^2}\right)$$

The posterior log-likelihood function $l(\mu, \sigma^2 \mid \xi)$ is re-defined as follows:

$$l(\mu,\sigma^2|\xi) = \log\left(\exp\left(-\frac{(\mu - \hat{\mu})^2}{2\hat{\sigma}^2}\right)\right) = -\frac{(\mu - \hat{\mu})^2}{2\hat{\sigma}^2}$$

Obviously, quadric function $l(\mu, \sigma^2 \mid \xi)$ gets maximal at $\mu = \hat{\mu}$ for each $\sigma^2$. In other words, we obtain equation 4.30 to calculate estimates $\hat{\mu}$ and $\hat{\sigma}^2$.

$$\begin{cases} \hat{\mu} = \widehat{\Theta} = \dfrac{\sigma^2\mu_0 + m\sigma_0^2\bar{X}}{\sigma^2 + m\sigma_0^2} \\ \hat{\sigma}^2 = \dfrac{\sigma^2\sigma_0^2}{\sigma^2 + m\sigma_0^2} \end{cases} \quad (4.30)$$

Where,

$$\bar{X} = \frac{1}{m}\sum_{i=1}^{m} X_i$$

Note, variance $\sigma^2$, prior mean $\mu_0$, and prior variance $\sigma_0^2$ in equation 4.30 are pre-defined (known). The estimate $\hat{\mu}$ is much more important than the estimate $\hat{\sigma}^2$ because given $\hat{\mu}$ the posterior log-likelihood function $l(\hat{\mu}, \sigma^2 \mid \xi)$ gets maximal for each $\sigma^2$.

Given estimates $\hat{\mu}$ and $\hat{\sigma}^2$, posterior density function (posterior distribution) of $\Theta = \mu$ is specified by equation 4.31.

$$P(\mu|D,\xi) = \mathcal{N}(\mu|D,\hat{\mu},\hat{\sigma}^2/m) = \frac{1}{\sqrt{2\pi\,\hat{\sigma}^2/m}} e^{-\frac{(\mu-\hat{\mu})^2}{2\hat{\sigma}^2/m}} \quad (4.31)$$

Note that $\xi = (\hat{\mu}, \hat{\sigma}^2/m)^T$. Obviously, posterior density function $P(\mu \mid D, \xi)$ distributes normally with mean $\hat{\mu}$ and variance $\frac{\hat{\sigma}^2}{m}$. The variance $\frac{\hat{\sigma}^2}{m}$ of posterior density function $P(\mu \mid D, \xi)$ is $m$ times smaller than the variance $\hat{\sigma}^2$ of updated density function mentioned later because $\mu$ is mean of $X$.

Because posterior density function $P(\mu \mid D, \xi)$ and prior density function $P(\mu \mid \xi)$ are conjugate probabilities, posterior density function $P(\mu \mid D, \xi)$ distributes normally. Updated probability $P(X \mid D)$ is now called updated density function specified equation 4.32.

$$P(X|D) = \mathcal{N}(X|D,\hat{\mu},\hat{\sigma}^2) = \frac{1}{\sqrt{2\pi\hat{\sigma}^2}} e^{-\frac{(X-\hat{\mu})^2}{2\hat{\sigma}^2}} \quad (4.32)$$

Obviously, updated density function $P(X \mid D)$ distributes normally with mean $\hat{\mu}$ and variance $\hat{\sigma}^2$.

The estimate $\hat{\mu}$ specified in equation 4.31 is theoretical mean of random variable (parameter) $\Theta = \mu$ and so we have:

$$\hat{\mu} = \int_{\mu} \mu P(\mu|D,\xi)\mathrm{d}\mu$$

In other words, $\hat{\mu} = \hat{\Theta}$ specified in equation 4.30 is Bayesian estimate of $\Theta = \mu$ under squared-error loss function.

Recall that the estimate $\hat{\mu}$ is much more important than the estimate $\hat{\sigma}^2$ because given $\hat{\mu}$ the posterior log-likelihood function $l(\hat{\mu}, \sigma^2 \mid \xi)$ gets maximal for each $\sigma^2$. Therefore, another estimate of $\sigma^2$ can be different from the estimate $\hat{\sigma}^2$ specified in equation 4.31. In fact, given $\hat{\mu}$ the posterior log-likelihood function $l(\mu, \sigma^2 \mid \xi)$ becomes $l(\sigma^2 \mid \xi)$ which is function of $\sigma^2$ as follows:

$$l(\sigma^2|\xi) = \log\big(P(D|\hat{\mu},\sigma^2)P(\hat{\mu}|\xi)\big) = \log\big(P(D|\Theta)\big) + \log\big(P(\hat{\mu}|\xi)\big)$$

$$= \log\left(\prod_{i=1}^{m} P(X_i|\hat{\mu},\sigma^2)\right) + \log\big(P(\hat{\mu}|\xi)\big)$$

$$= \log\left(\prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(X_i-\hat{\mu})^2}{2\sigma^2}}\right) + \log\left(\frac{1}{\sqrt{2\pi\sigma_0^2}} e^{-\frac{(\hat{\mu}-\mu_0)^2}{2\sigma_0^2}}\right)$$

$$= -\frac{m}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{m}(X_i-\hat{\mu})^2 - \frac{1}{2}\log(2\pi\sigma_0^2) - \frac{1}{2\sigma_0^2}(\hat{\mu}-\mu_0)^2 \blacksquare$$

The first-order derivative of posterior log-likelihood function $l(\sigma^2 \mid \xi)$ with regard to $\sigma^2$ is:

$$\frac{\partial l(\sigma^2|\xi)}{\partial\sigma^2} = \frac{\sum_{i=1}^{m}(X_i-\hat{\mu})^2}{2(\sigma^2)^2} - \frac{m}{2\sigma^2}$$

Let $(\sigma^2)^*$ be another estimate of $\sigma^2$. The estimate $(\sigma^2)^*$ is solution of the equation formed by setting the first-order derivative $\frac{\partial l(\sigma^2|\xi)}{\partial\sigma^2}$ to be zero as follows:

$$\frac{\sum_{i=1}^m (X_i-\hat{\mu})^2}{2(\sigma^2)^2}-\frac{m}{2\sigma^2}\Rightarrow\frac{\sum_{i=1}^m (X_i-\hat{\mu})^2}{\sigma^2}-m=0\Rightarrow\sigma^2=\frac{1}{m}\sum_{i=1}^m (X_i-\hat{\mu})^2$$

In short, equation 4.33 specifies $(\sigma^2)^*$ which is another estimate of $\sigma^2$ with note that $\hat{\mu}$ is the estimate of $\mu$ specified in equation 4.30.

$$(\sigma^2)^*=\frac{1}{m}\sum_{i=1}^m (X_i-\hat{\mu})^2 \quad (4.33)$$

I think that the estimate $(\sigma^2)^*$ is more precise than the estimate $\hat{\sigma}^2$ because $(\sigma^2)^*$ concerns observations $X_i$. Given $(\sigma^2)^*$, posterior density function (equation 4.31) and updated density function (equation 4.32) are re-written as follows:

$$\begin{gathered}P(\mu|D,\xi)=\mathcal{N}(\mu|D,\hat{\mu},(\sigma^2)^*/m)=\frac{1}{\sqrt{2\pi\,(\sigma^2)^*/m}}e^{-\frac{(\mu-\hat{\mu})^2}{2(\sigma^2)^*/m}}\\ P(X|D)=\mathcal{N}(X|D,\hat{\mu},(\sigma^2)^*)=\frac{1}{\sqrt{2\pi(\sigma^2)^*}}e^{-\frac{(X-\hat{\mu})^2}{2(\sigma^2)^*}}\end{gathered} \quad (4.34)$$

Basic concepts of Bayesian learning were introduced. Sub-sections 4.1 and 4.2 mention how to learn CPTs of BN in which sub-section 4.1 focuses on parameter learning in complete data whereas sub-section 4.2 focuses on parameter learning in incomplete data. In sub-sections 4.1 and 4.2, nodes in BN are binary random variables and data sample is binomial sample. Recall that the report focuses on discrete BN.

## 4.1. Parameter learning with binomial complete data

Given a discrete BN ($G$, $P$) satisfies Markov condition where both structure $G$ and parameter $P$ are known with note that $G$ is a DAG and $P$ is a joint probability distribution. Moreover, $P$ is formulated from CPTs, which means that $P$ is product of conditional probabilities of nodes given their parents according to theorem 2.1.2 (Neapolitan, 2003, p. 37), parameter learning here aims to improves CPTs from binomial complete data.

Suppose there is one binary variable $X$ in BN and probability distribution of $X$ is considered as relative frequency having values in space [0, 1] which is the range of variable $F$. A parameter $F$ (whose space is [0, 1], of course) is added to each variable $X$, which acts as the parent of $X$ and has a beta density function $\beta(F;\text{ a, b})$, so as to:

$$P(X=1\mid F)=F\text{, where }F\text{ has beta density function }\beta(F;a,b) \quad (4.1.1)$$

Please pay attention to equation 4.1.1, $P(X=1 \mid F) = F$ implicating that $F$ represents relative frequency of $X$ (Neapolitan, 2003, p. 301) because it is the key of learning CPT based on beta density function. Variable $X$ and parameter $F$ constitute a simple network which is referred as augmented BN (Neapolitan, 2003, p. 324). Figure 4.1.1 shows the simplest augmented BN. We use binomial sample to learn BN and $F$ is essentially the parameter Θ of binomial sampling, $F$ = Θ. Because parameter is considered as random variable in Bayesian approach, $F$ is called *augmented variable* as a convention.

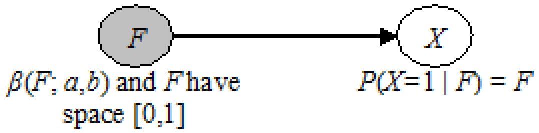


**Figure 4.1.1.** The simple (binomial) augmented BN with only one hypothesis node *X*

The augmented BN is often denoted as a triple ($G$, $F^{(G)}$, $\beta^{(G)}$) whereas the BN is denoted as a pair ($G$, $P$). As a convention, ($G$, $F^{(G)}$, $\rho^{(G)}$) is called *augmented BN* of ($G$, $P$) and ($G$, $P$) is called *embedded BN* of ($G$, $F^{(G)}$, $\beta^{(G)}$). If $\rho$ is beta distribution, we denote ($G$, $F^{(G)}$, $\beta^{(G)}$) as augmented BN. Moreover, we can denote ($G$, $F$, $\beta$) and ($G$, $F$, $\rho$) if $G$ is implied.

The probability $P(X = 1)$ which is parameter of BN is really prior predictive probability and so we have a simple but effective equation 4.1.2 to compute $P(X = 1)$ as follows:

$$P(X=1)=E(F)=\frac{a}{N} \qquad (4.1.2)$$

Following is the proof of equation 4.1.2.

$$P(X=1)=\int_0^1 P(X=1|F)\beta(F)\mathrm{d}F=\int_0^1 F\beta(F)\mathrm{d}F=E(F)=\frac{a}{N}\blacksquare$$

Note, $P(X=1)$ is CPT of $X$. Please refer to equation 4.14 to know how to calculate mean of beta distribution. Please pay attention to equation 4.1.2, it is the most essential equation used in parameter learning. The equation 4.1.2 is corollary 6.1 in (Neapolitan, 2003, p. 302).

The ultimate purpose of Bayesian inference is to consolidate a hypothesis (namely, variable) by collecting evidences. Suppose we perform $M$ trials of a random process, the outcome of $u^{th}$ trial is denoted $X^{(u)}$ considered as evidence variable whose probability $P(X^{(u)} = 1 \mid F) = F$. So, all $X^{(u)}$ are conditionally dependent on $F$. The probability of variable $X$, $P(X=1)$ is learned by these evidences. Note that evidence $X^{(u)}$ is considered as random variable like $X$.

We denote the vector of all evidences as $\mathcal{D} = (X^{(1)}, X^{(2)},\ldots, X^{(m)})$ which is also called the sample of size $m$. Hence, $\mathcal{D}$ is known as a sample or an evidence vector and we often implicate $\mathcal{D}$ as a collection of evidences. Given this sample, $\beta(F)$ is called prior density function, and $P(X^{(u)} = 1) = a/N$ (due to equation 4.1.2) is called prior probability of $X^{(u)}$. It is necessary to determine posterior density function $\beta(F|\mathcal{D})$ and updated probability of $X$, namely $P(X|\mathcal{D})$. *The nature of this process is the parameter learning* which aims to determine CPTs that are parameters of discrete BN with note that such CPTs essentially are updated probabilities $P(X|\mathcal{D})$. Note, $P(X|\mathcal{D})$ can be referred as $P(X^{(m+1)} \mid \mathcal{D})$. Figure 4.1.2 depicts this sample $\mathcal{D} = (X^{(1)}, X^{(2)},\ldots, X^{(m)})$.

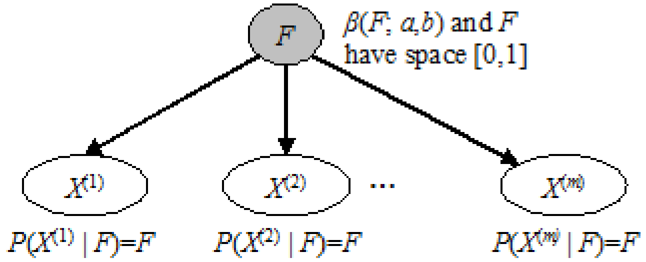


**Figure 4.1.2.** The binomial sample $\mathcal{D}=(X^{(1)}, X^{(2)},\ldots, X^{(m)})$ of size $m$

We survey firstly the case of binomial sample. Thus, $\mathcal{D}$ having binomial distribution is called binomial sample and the network in figure 4.1.1 becomes a binomial augmented BN. Then, suppose $s$ is the number of all evidences $X^{(i)}$ which have value 1 (success), otherwise, $t$ is the number of all evidences $X^{(j)}$ which have value 0 (failed). Of course, $s + t = M$. Note that $s$ and $t$ are often called counters or count numbers.

**Computing posterior density function and updated probability**

Now, we need to compute posterior density function $\beta(F|\mathcal{D})$ and updated probability $P(X=1|\mathcal{D})$. It is essential to determine probability distribution of $X$. Fortunately, $\beta(F|\mathcal{D})$ and $P(X=1|\mathcal{D})$ are already determined by equations 4.15 and 4.16 when $F = \Theta$ and $P(X=1|\mathcal{D}) = P(X_{n+1}=1|\mathcal{D})$. For convenience, we replicate equations 4.15 and 4.16 as equations 4.1.3 and 4.1.4, respectively.

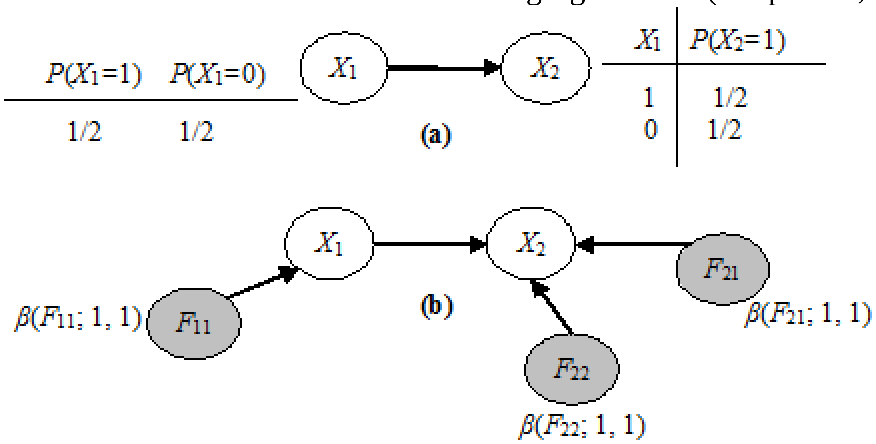
$$\beta(F|\mathcal{D}) = \beta(F; a+s, b+t) \tag{4.1.3}$$

$$P(X=1|\mathcal{D}) = E(F|\mathcal{D}) = \frac{a+s}{N+M} \tag{4.1.4}$$

From equation 4.1.4, $P(X=1|\ \mathcal{D})$ representing updated CPT of $X$ is an estimate of $F$ under squared-error loss function. Equation 4.1.4 is theorem 6.4 (Neapolitan, 2003, p. 309). In general, you should merely remember equations 4.1.2 and 4.1.4 to calculate probability of $X$ and updated probability of $X$, respectively. Essentially, equations 4.17 or 4.1.4 is special case of equation 4.6 in case of binomial sampling and beta prior distribution, which is used to estimate $F$ under squared-error loss function.

**Expanding augmented BN with more than one hypothesis node**

Suppose we have a BN with two binary random variables and there is conditional dependence assertion between these nodes. Note, a BN having more than one hypothesis variable is known as multi-node BN. See the networks and CPTs in following figure 4.1.3 (Neapolitan, 2003, p. 329):



| $P(X_1=1)$ | $P(X_1=0)$ |
|---|---|
| 1/2 | 1/2 |

| $X_1$ | $P(X_2=1)$ |
|---|---|
| 1 | 1/2 |
| 0 | 1/2 |

**Figure 4.1.3.** BN (a) and complex augmented BN (b)

In figure 4.1.3, the BN (a) having no attached augmented variable is also called original BN or trust BN, from which augmented BN (b) is derived by the way: for every node (variable) $X_i$, we add parameter parent nodes to $X_i$, obeying two principles below:

1. If $X_i$ has no parent (not conditionally dependent on any others, $X_i$ is a root), we add only one augmented variable denoted $F_{i1}$ having probability density function $\beta(F_{i1}; a_{i1}, b_{i1})$ so as to $P(X_i=1|F_{i1}) = F_{i1}$.
2. If $X_i$ has a set of $p_i$ parent nodes and each parent node is binary, we add a set of $q_i=2p_i$ parameter variables $\{F_{i1}, F_{i2},\ldots, F_{iq_i}\}$ which, in turn, correspond to instances of parent nodes of $X_i$, namely $\{PA_{i1}, PA_{i2}, PA_{i3},\ldots, PA_{iq_i}\}$ where each $PA_{ij}$ is an instance of a parent node of $X_i$ with note that each binary parent node has two instances (0 and 1, for example). For convenience, each $PA_{ij}$ is called a parent instance of $X_i$ and we let $PA_i=\{PA_{i1}, PA_{i2}, PA_{i3},\ldots, PA_{iq_i}\}$ be vector or collection of parent instances of $X_i$. We also let $F_i=\{F_{i1}, F_{i2},\ldots, F_{iq_i}\}$ be respective vector or collection of augmented variables $F_{i1}$ (s) attached to $X_i$. Now in a given augmented BN $(G, F^{(G)}, \beta^{(G)})$, $F$ is a set of all $F_i$ (s), $F = \{F_1, F_2,\ldots, F_n\}$ in which each $F_i$ is a vector of $F_{ij}$ (s) and in turn each $F_{ij}$ is a root node. It is conventional that each $X_i$ has $q_i$ parent instances ($q_i \geq 0$); in other words, $q_i$ denotes the size of $PA_i$ and the size of $F_i$. For example, in figure 4.1.3, node $X_2$ has one parent node $X_1$, which causes that $X_2$

has two parent instances represented by two augmented variables $F_{21}$ and $F_{22}$. Additionally, $F_{21}$ ($F_{22}$) and its beta density function specify conditional probabilities of $X_2$ given $X_1 = 1$ ($X_1 = 0$) because parent node $X_1$ is binary. We have equation 4.1.5 for connecting CPT of variable $X_i$ with beta density function of augmented variable $F_i$.

$$P\left(X_i = 1 \middle| PA_{ij}, F_{i1}, F_{i2}, \dots, F_{ij}, \dots, F_{iq_i}\right) = P\left(X_i = 1 \middle| PA_{ij}, F_{ij}\right) = F_{ij} \tag{4.1.5}$$

Equation 4.1.5 is an extension of equation 4.1.1 in multi-node BN and equation 4.1.5 degenerates to equation 4.1.1 if $X_i$ has no parent. Note that the beta density function of $F_{ij}$ is $\beta(F_{ij}; a_{ij}, b_{ij})$ and of course, in figure 4.1.3, we have $a_{11}$=1, $b_{11}$=1, $a_{21}$=1, $b_{21}$=1, $a_{22}$=1, $b_{22}$=1.

Beta density function for each $F_{ij}$ is specified in equation 4.1.6 as follows:

$$\beta\left(F_{ij}\right) = \beta\left(F_{ij} \middle| a_{ij}, b_{ij}\right) = \frac{\Gamma\left(N_{ij}\right)}{\Gamma\left(a_{ij}\right)\Gamma\left(b_{ij}\right)} {F_{ij}}^{a_{ij}-1}\left(1 - F_{ij}\right)^{b_{ij}-1} \tag{4.1.6}$$

Where $N_{ij} = a_{ij} + b_{ij}$. Given augmented BN ($G$, $F^{(G)}$, $\beta^{(G)}$), notation $\beta$ implies set of all $\beta(F_{ij})$ which in turn implies set of all ($a_{ij}$, $b_{ij}$). Note that equations 4.12 and 4.1.6 have the same meaning for representing beta function except that equation 4.1.6 is used in multi-node BN. Variables $F_{ij}$ (s) attached to the same $X_i$ have no parent and are mutually independent, so, it is very easy to compute the joint beta density function $\beta(F_{i1}, F_{i2},\dots, F_{iq_i})$ with regard to node $X_i$ as follows:

$$\beta(F_i) = \beta\left(F_{i1}, F_{i2}, \dots, F_{ic_i}\right) = \beta(F_{i1})\beta(F_{i2}) \dots \beta\left(F_{ic_i}\right) = \prod_{j=1}^{q_i} \beta\left(F_{ij}\right) \tag{4.1.7}$$

Besides the local parameter independence expressed in equation 4.1.7, we have global parameter independence if reviewing all variables $X_i$ (s) with note that all respective $F_{ij}$ (s) over entire augmented BN are mutually independent. Equation 4.1.8 expresses the global parameter independence of all $F_{ij}$ (s).

$$\begin{aligned}\beta(F_1, F_2, \dots, F_i, \dots, F_n) &= \beta\begin{pmatrix} F_{11}, F_{12}, \dots, F_{1q_1}, F_{21}, F_{22}, \dots, F_{2q_2}, \dots, \\ F_{i1}, F_{i2}, \dots, F_{iq_i}, \dots, F_{n1}, F_{n2}, \dots, F_{nq_n} \end{pmatrix} \\ &= \prod_{i=1}^{n} \beta\left(F_{i1}, F_{i2}, \dots, F_{iq_i}\right) = \prod_{i=1}^{n}\prod_{j=1}^{q_i} \beta\left(F_{ij}\right)\end{aligned} \tag{4.1.8}$$

Concepts “local parameter independence” and “global parameter independence” are defined in (Neapolitan, 2003, p. 333).

All variables $X_i$ and their augmented variables form the complex augmented BN representing the trust BN in figure 4.1.3. In the trust BN, the conditional probability of variable $X_i$ with respect to its parent instance $PA_{ij}$, in other words, the $ij^{th}$ conditional distribution is the expected value of $F_{ij}$ as below:

$$P\left(X_i = 1 \middle| PA_{ij}\right) = E\left(F_{ij}\right) = \frac{a_{ij}}{N_{ij}} \tag{4.1.9}$$

Equation 4.1.9 is extension of equation 4.1.2 when variable $X_i$ has parent and both equations express prior probability of variable $X_i$. Following is proof of equation 4.1.9.

$$\begin{aligned}&P\left(X_i = 1 \middle| PA_{ij}\right) \\ &= \int_0^1 \dots \int_0^1 P\left(X_i = 1 \middle| PA_{ij}, F_{i1}, \dots, F_{ij}, \dots, F_{iq_i}\right)\beta\left(F_{i1}, \dots, F_{ij}, \dots, F_{iq_i}\right) \\ &\qquad \mathrm{d}F_{i1} \dots \mathrm{d}F_{ij} \dots \mathrm{d}F_{iq_i}\end{aligned}$$

$$= \int_0^1 \dots \int_0^1 P\left(X_i = 1 \middle| PA_{ij}, F_{i1}, \dots, F_{ij}, \dots, F_{iq_i}\right)\beta(F_{i1}) \dots \beta\left(F_{ij}\right) \dots \beta\left(F_{iq_i}\right)$$
$$\mathrm{d}F_{i1} \dots \mathrm{d}F_{ij} \dots \mathrm{d}F_{iq_i}$$

(due to local parameter independence specified in equation 4.1.7 when *F<sub></sub>* $F_{ij}$ (s) are mutually independent)

$$= \int_0^1 \dots \int_0^1 F_{ij}\beta(F_{i1}) \dots \beta\left(F_{ij}\right) \dots \beta\left(F_{iq_i}\right)\mathrm{d}F_{i1} \dots \mathrm{d}F_{ij} \dots \mathrm{d}F_{iq_i}$$

$$\left(\text{due to } P\left(X_i = 1 \middle| PA_{ij}, F_{i1}, \dots, F_{ij}, \dots, F_{iq_i}\right) = F_{ij} \text{ specified in equation 4.1.5}\right)$$

$$= \left(\int_0^1 \beta(F_{i1})\mathrm{d}F_{i1}\right) * \dots * \left(\int_0^1 F_{ij}\beta\left(F_{ij}\right)\mathrm{d}F_{ij}\right) * \dots * \left(\int_0^1 \beta\left(F_{iq_i}\right)\mathrm{d}F_{iq_i}\right)$$

$$= 1 * \dots * \left(\int_0^1 F_{ij}\beta\left(F_{ij}\right)\mathrm{d}F_{ij}\right) * \dots * 1$$

$$= \int_0^1 F_{ij}\beta\left(F_{ij}\right)\mathrm{d}F_{ij} = E\left(F_{ij}\right) = \frac{a_{ij}}{N_{ij}} \blacksquare$$

Equation 4.1.9 is theorem 6.7 proved by the similar way in (Neapolitan, 2003, pp. 334-335) to which I referred.

**Example 4.1.1.** For illustrating equations 4.1.5 and 4.1.9, recall that variables $F_{ij}$ (s) and their beta density functions $\beta(F_{ij})$ (s) specify conditional probabilities of $X_i$ (s) as in figure 4.1.3, and so, the CPTs in figure 4.1.3 is interpreted in detailed as follows:

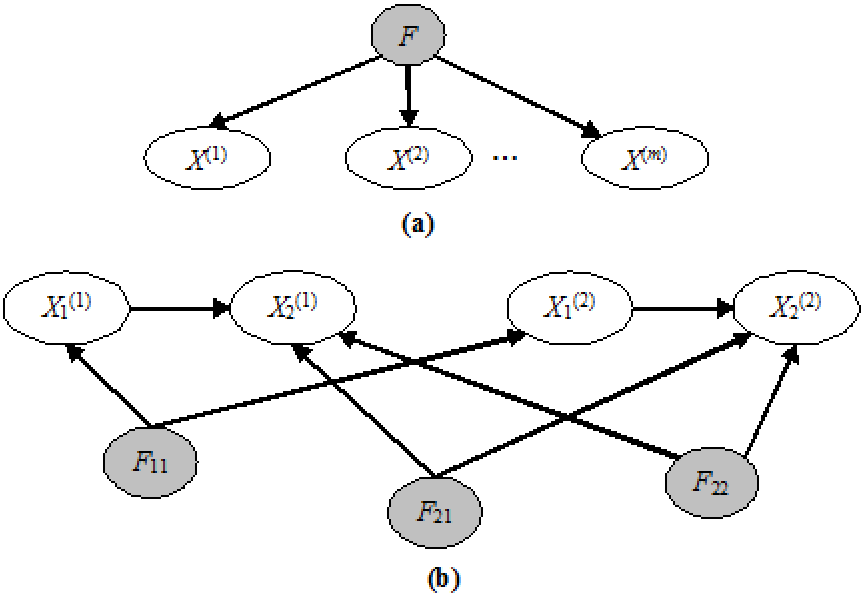
$$P(X_1 = 1|F_{11}) = F_{11} \Rightarrow P(X_1 = 1) = E(F_{11}) = \frac{1}{1+1} = \frac{1}{2}$$
$$P(X_2 = 1|X_1 = 1, F_{21}) = F_{21} \Rightarrow P(X_2 = 1|X_1 = 1) = E(F_{21}) = \frac{1}{1+1} = \frac{1}{2}$$
$$P(X_2 = 1|X_1 = 0, F_{22}) = F_{22} \Rightarrow P(X_2 = 1|X_1 = 0) = E(F_{22}) = \frac{1}{1+1} = \frac{1}{2}$$

Note that inverted probabilities in CPTs such as $P(X_1=0)$, $P(X_2=0|X_1=1)$ and $P(X_2=0|X_1=0)$ are not mentioned because $X_i$ (s) are binary variables and so, $P(X_1=0) = 1 - P(X_1=1) = 1/2$, $P(X_2=0|X_1=1) = 1 - P(X_2=1|X_1=1) = 1/2$ and $P(X_2=0|X_1=0) = 1 - P(X_2=1|X_1=0) = 1/2$■

Suppose we perform *m* trials of random process, the outcome of *u*th trial which is BN like figure 4.1.3 is represented as a random vector $X^{(u)}$ containing all evidence variables in network. Vector $X^{(u)}$ is also called the *u*th *evidence* (vector) of entire BN. Suppose $X^{(u)}$ has *n* components or partial evidences $X_i^{(u)}$ when BN has *n* nodes; in figure 4.1.3, *n* = 2. Note that evidence $X_i^{(u)}$ is considered as random variable like $X_i$.

$$X^{(u)} = \begin{pmatrix} X_1^{(u)} \\ X_2^{(u)} \\ \vdots \\ X_n^{(u)} \end{pmatrix}$$

It is easy to recognize that each component $X_i^{(u)}$ represents the *u*th evidence of node $X_i$ in the BN. The *m* trials constitute the sample of size *m* which is the set of random vectors denoted as $\mathcal{D}=\{X^{(1)}, X^{(2)},\dots, X^{(m)}\}$. $\mathcal{D}$ is also called evidence matrix, evidence sample, training data, or evidences, in brief. We only review the case of binomial sample; it means that $\mathcal{D}$ is the binomial BN sample of size *m*. For example, this sample corresponding to the network in figure 4.1.3 is depicted by figure 4.1.4 as below (Neapolitan, 2003, p. 337):



(a)

(b)

**Figure 4.1.4.** Expanded binomial augmented BN sample of size *m*

After *m* trials are performed, the augmented BN are updated and so, augmented variables' density functions and hypothesis variables' conditional probabilities are changed. We need to compute posterior density function $\beta(F_{ij}|\mathcal{D})$ of each augmented variable $F_{ij}$ and updated condition probability $P(X_i=1|\ PA_{ij}, \mathcal{D})$ of each variable $X_i$. Note that evidence vectors $X^{(u)}$ (s) are mutually independent given all $F_{ij}$ (s). It is easy to infer that given fixed *i*, all evidences $X_i^{(u)}$ corresponding to variable $X_i$ are mutually independent. Based on binomial trials and mentioned mutual independence, equation 4.1.10 is used for calculating probability of evidences corresponding to variable $X_i$ over *m* trials as follows:

$$P\left(X_i^{(1)}, X_i^{(2)}, \dots, X_i^{(m)} \middle| PA_i, F_i\right) = \prod_{u=1}^{m} P\left(X_i^{(u)} \middle| PA_i, F_i\right) = \prod_{j=1}^{q_i} \left(F_{ij}\right)^{s_{ij}} \left(1 - F_{ij}\right)^{t_{ij}} \quad (4.1.10)$$

Where,

- Number $q_i$ is the number of parent instances of $X_i$. In binary case, each $X_i^{(u)}$ 's parent node has two instances/values, namely, 0 and 1.
- Counter $s_{ij}$, respective to $F_{ij}$, is the number of all evidences among *m* trials such that variable $X_i$ = 1 and $PA_{ij}$ = 1. Counter $t_{ij}$, respective to $F_{ij}$, is the number of all evidences among *m* trials such that variable $X_i$ = 1 and $PA_{ij}$ = 0. Note that $s_{ij}$ and $t_{ij}$ are often called *counters* or count numbers.
- $PA_i=\{PA_{i1}, PA_{i2}, PA_{i3}, \dots, PA_{iq_i}\}$ is the vector of parent instances of $X_i$ and $F_i = \{F_{i1}, F_{i2}, \dots, F_{iq_i}\}$ is the respective vector of variables $F_{i1}$ (s) attached to $X_i$.

Please see equation 4.9 to understand equation 4.1.10. From equation 4.1.10, it is easy to compute likelihood function $P(\mathcal{D}|F_1, F_2, \dots, F_n)$ of evidence sample $\mathcal{D}$ given *n* vectors $F_i$ (s) with assumption that BN has *n* variables $X_i$ (s) as follows:

$$P(\mathcal{D}|F_1, F_2, \dots, F_n) = P\left(X^{(1)}, X^{(2)}, \dots, X^{(m)} \middle| F_1, F_2, \dots, F_n\right) = \prod_{u=1}^{m} P\left(X^{(u)} \middle| F_1, F_2, \dots, F_n\right)$$

(because evidence vectors $X^{(u)}$ (s) are mutually independent)

$$=\prod_{u=1}^{m}\frac{P\left(X^{(u)},F_1,F_2,\dots,F_n\right)}{P(F_1,F_2,\dots,F_n)}$$

(due to Bayes' rule specified in equation 1.1)

$$=\prod_{u=1}^{m}\frac{P\left(X_1^{(u)},X_2^{(u)},\dots,X_n^{(u)},F_1,F_2,\dots,F_n\right)}{P(F_1,F_2,\dots,F_n)}$$

$$=\prod_{u=1}^{m}\frac{P\left(X_1^{(u)},X_2^{(u)},\dots,X_n^{(u)}\middle|F_1,F_2,\dots,F_n\right)P(F_1,F_2,\dots,F_n)}{P(F_1,F_2,\dots,F_n)}$$

(applying multiplication rule specified by equation 1.3 into the numerator)

$$=\prod_{u=1}^{m}P\left(X_1^{(u)},X_2^{(u)},\dots,X_n^{(u)}\middle|F_1,F_2,\dots,F_n\right)$$

$$=\prod_{u=1}^{m}\prod_{i=1}^{n}P\left(X_i^{(u)}\middle|PA_i,F_i\right)$$

(because $X_i^{(u)}$ (s) are mutually independent given $F_i$ (s) and each $X_i$ depends only on $PA_i$ and $F_i$)

$$=\prod_{i=1}^{n}\prod_{u=1}^{m}P\left(X_i^{(u)}\middle|PA_i,F_i\right)=\prod_{i=1}^{n}\prod_{j=1}^{q_i}\left(F_{ij}\right)^{s_{ij}}\left(1-F_{ij}\right)^{t_{ij}}$$

$$\left(\text{due to equation 4.1.10: }\prod_{u=1}^{m}P\left(X_i^{(u)}\middle|PA_i,F_i\right)=\prod_{j=1}^{q_i}\left(F_{ij}\right)^{s_{ij}}\left(1-F_{ij}\right)^{t_{ij}}\right)\blacksquare$$

In brief, we have equation 4.1.11 for calculating likelihood function $P(\mathcal{D}|F_1, F_2,\dots, F_n)$ of evidence sample $\mathcal{D}$ given $n$ vectors $F_i$ (s).

$$P(\mathcal{D}|F_1,F_2,\dots,F_n)=\prod_{i=1}^{n}\prod_{j=1}^{q_i}\left(F_{ij}\right)^{s_{ij}}\left(1-F_{ij}\right)^{t_{ij}}\quad(4.1.11)$$

The equation 4.1.11 is lemma 6.8 proved by similar way in (Neapolitan, 2003, pp. 338-339) to which I referred. It is necessary to calculate marginal probability $P(\mathcal{D})$ of evidence sample $\mathcal{D}$, we have:

$$P(\mathcal{D})=P\left(X^{(1)},X^{(2)},\dots,X^{(m)}\right)=\prod_{u=1}^{m}P\left(X^{(u)}\right)=\prod_{u=1}^{m}P\left(X_1^{(u)},X_2^{(u)},\dots,X_n^{(u)}\right)$$

(due evidence vectors $X^{(u)}$ (s) are independent)

$$=\prod_{u=1}^{m}\int_{F_1}\dots\int_{F_n}P\left(X_1^{(u)},X_2^{(u)},\dots,X_n^{(u)}\middle|F_1,F_2,\dots,F_n\right)\beta(F_1,F_2,\dots,F_n)\,\mathrm{d}F_1\mathrm{d}F_2\dots\mathrm{d}F_n$$

(due to total probability rule in continuous case, please see equation 1.5)

$$=\prod_{u=1}^{m}\int_{F_1}\dots\int_{F_n}\prod_{i=1}^{n}P\left(X_i^{(u)}\middle|PA_i,F_i\right)\prod_{i=1}^{n}\beta(F_i)\,\mathrm{d}F_1\mathrm{d}F_2\dots\mathrm{d}F_n$$

(Because $X_i^{(u)}$ (s) are mutually independent given $F_i$ (s) and each $X_i$ depends only on $PA_i$ and $F_i$. Moreover, all $F_i$ (s) are mutually independent)

$$=\prod_{u=1}^{m}\int_{F_1}\dots\int_{F_n}\left(\prod_{i=1}^{n}P\left(X_i^{(u)}\middle|PA_i,F_i\right)\beta(F_i)\right)\mathrm{d}F_1\mathrm{d}F_2\dots\mathrm{d}F_n$$

$$=\prod_{u=1}^{m}\prod_{i=1}^{n}\int_{F_i}P\left(X_i^{(u)}\middle|PA_i,F_i\right)\beta(F_i)\mathrm{d}F_i=\prod_{i=1}^{n}\prod_{u=1}^{m}\int_{F_i}P\left(X_i^{(u)}\middle|PA_i,F_i\right)\beta(F_i)\mathrm{d}F_i$$

$$=\prod_{i=1}^{n}\prod_{j=1}^{q_i}\int_0^1\left(F_{ij}\right)^{s_{ij}}\left(1-F_{ij}\right)^{t_{ij}}\beta\left(F_{ij}\right)\mathrm{d}F_{ij}$$

$$\left(\text{due to binomial trials: }\prod_{u=1}^{m}\int_{F_i}P\left(X_i^{(u)}\middle|PA_i,F_i\right)\beta(F_i)\mathrm{d}F_i=\prod_{j=1}^{q_i}\int_0^1\left(F_{ij}\right)^{s_{ij}}\left(1-F_{ij}\right)^{t_{ij}}\beta\left(F_{ij}\right)\mathrm{d}F_{ij}\right)$$

$$=\prod_{i=1}^{n}\prod_{j=1}^{q_i}E\left(\left(F_{ij}\right)^{s_{ij}}\left(1-F_{ij}\right)^{t_{ij}}\right)\blacksquare$$

In brief, we have following equation which is theorem 6.11 in (Neapolitan, 2003, p. 343) for determining marginal probability $P(\mathcal{D})$ of evidence sample $\mathcal{D}$ as product of expectations of binomial trials.

$$P(\mathcal{D})=\prod_{i=1}^{n}\prod_{j=1}^{q_i}E\left(\left(F_{ij}\right)^{s_{ij}}\left(1-F_{ij}\right)^{t_{ij}}\right)$$

There is the question "how to determine $E\left(\left(F_{ij}\right)^{s_{ij}}\left(1-F_{ij}\right)^{t_{ij}}\right)$ in equation above" and so we have equation 4.1.12 for calculating both this expectation and $P(\mathcal{D})$ by referring to equation 4.15 when all $F_{ij}$ are independent, as follows:

$$E\left(\left(F_{ij}\right)^{s_{ij}}\left(1-F_{ij}\right)^{t_{ij}}\right)=\frac{\Gamma\left(N_{ij}\right)}{\Gamma\left(N_{ij}+M_{ij}\right)}\frac{\Gamma\left(a_{ij}+s_{ij}\right)\Gamma\left(b_{ij}+t_{ij}\right)}{\Gamma\left(a_{ij}\right)\Gamma\left(b_{ij}\right)}$$

$$P(\mathcal{D})=\prod_{i=1}^{n}\prod_{j=1}^{q_i}E\left(\left(F_{ij}\right)^{s_{ij}}\left(1-F_{ij}\right)^{t_{ij}}\right)=\prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{\Gamma\left(N_{ij}\right)}{\Gamma\left(N_{ij}+M_{ij}\right)}\frac{\Gamma\left(a_{ij}+s_{ij}\right)\Gamma\left(b_{ij}+t_{ij}\right)}{\Gamma\left(a_{ij}\right)\Gamma\left(b_{ij}\right)}\quad(4.1.12)$$

Where $N_{ij}=a_{ij}+b_{ij}$ and $M_{ij}=s_{ij}+t_{ij}$. When both likelihood function $P(\mathcal{D}|F_1, F_2,\dots, F_n)$ and marginal probability $P(\mathcal{D})$ for evidences are determined, it is easy to update the probability of $X_i$. That is the main subject of parameter learning.

**Computing posterior density function and updated probability in multi-node BN**

Now, we need to compute posterior density function $\beta(F_{ij}|\mathcal{D})$ and updated probability $P(X_i=1|PA_{ij}, \mathcal{D})$ for each variable $X_i$ in BN. In fact, we have:

$$\beta\left(F_{ij}\middle|\mathcal{D}\right)=\frac{P\left(\mathcal{D}\middle|F_{ij}\right)\beta\left(F_{ij}\right)}{P(\mathcal{D})}$$

(due to Bayes' rule specified in equation 1.1)

$$=\frac{\left(\int_0^1\dots\int_0^1 P(\mathcal{D}|F_1,F_2,\dots,F_n)\prod_{kl\neq ij}\beta(F_{kl})\mathrm{d}F_{kl}\right)\beta\left(F_{ij}\right)}{P(\mathcal{D})}$$

(Due to total probability rule in continuous case, specified by equation 1.5. Note that $F_i = \{F_{i1}, F_{i2},\dots, F_{iq_i}\}$)

$$=\frac{\left(\int_0^1\dots\int_0^1\left(\prod_{u=1}^{n}\prod_{v=1}^{q_u}(F_{uv})^{s_{uv}}(1-F_{uv})^{t_{uv}}\right)\left(\prod_{kl\neq ij}\beta(F_{kl})\mathrm{d}F_{kl}\right)\right)\beta\left(F_{ij}\right)}{P(\mathcal{D})}$$

(due to equation 4.1.11)

$$=\frac{\left(F_{ij}\right)^{s_{ij}}\left(1-F_{ij}\right)^{t_{ij}}\left(\int_0^1\dots\int_0^1\prod_{kl\neq ij}(F_{kl})^{s_{kl}}(1-F_{kl})^{t_{kl}}\beta(F_{kl})\mathrm{d}F_{kl}\right)\beta\left(F_{ij}\right)}{P(\mathcal{D})}$$

$$=\frac{\left(F_{ij}\right)^{s_{ij}}\left(1-F_{ij}\right)^{t_{ij}}\left(\prod_{kl\neq ij}\int_0^1(F_{kl})^{s_{kl}}(1-F_{kl})^{t_{kl}}\beta(F_{kl})\mathrm{d}F_{kl}\right)\beta\left(F_{ij}\right)}{P(\mathcal{D})}$$

$$=\frac{\left(F_{ij}\right)^{s_{ij}}\left(1-F_{ij}\right)^{t_{ij}}\left(\prod_{kl\neq ij}E((F_{kl})^{s_{kl}}(1-F_{kl})^{t_{kl}})\right)\beta\left(F_{ij}\right)}{\prod_{k=1}^{n}\prod_{l=1}^{q_k}E((F_{kl})^{s_{kl}}(1-F_{kl})^{t_{kl}})}$$

(applying equation 4.1.12 into denominator)

$$=\frac{\left(F_{ij}\right)^{s_{ij}}\left(1-F_{ij}\right)^{t_{ij}}\left(\prod_{kl\neq ij}E((F_{kl})^{s_{kl}}(1-F_{kl})^{t_{kl}})\right)\beta\left(F_{ij}\right)}{\prod_{kl}E((F_{kl})^{s_{kl}}(1-F_{kl})^{t_{kl}})}$$

$$=\frac{\left(F_{ij}\right)^{s_{ij}}\left(1-F_{ij}\right)^{t_{ij}}\beta\left(F_{ij}\right)}{E\left(\left(F_{ij}\right)^{s_{ij}}\left(1-F_{ij}\right)^{t_{ij}}\right)}$$

$$=\frac{\left(F_{ij}\right)^{s_{ij}}\left(1-F_{ij}\right)^{t_{ij}}\dfrac{\Gamma\left(N_{ij}\right)}{\Gamma\left(a_{ij}\right)\Gamma\left(b_{ij}\right)}\left(F_{ij}\right)^{a_{ij}-1}\left(1-F_{ij}\right)^{b_{ij}-1}}{\dfrac{\Gamma\left(N_{ij}\right)}{\Gamma\left(N_{ij}+M_{ij}\right)}\dfrac{\Gamma\left(a_{ij}+s_{ij}\right)\Gamma\left(b_{ij}+t_{ij}\right)}{\Gamma\left(a_{ij}\right)\Gamma\left(b_{ij}\right)}}$$

(applying definition of beta density function specified by equation 4.12 into numerator and applying equation 4.1.12 into denominator, note that $N_{ij} = a_{ij} + b_{ij}$ and $M_{ij} = s_{ij} + t_{ij}$)

$$=\frac{\Gamma\left(N_{ij}+M_{ij}\right)}{\Gamma\left(a_{ij}+s_{ij}\right)\Gamma\left(b_{ij}+t_{ij}\right)}\left(F_{ij}\right)^{a_{ij}+s_{ij}-1}\left(1-F_{ij}\right)^{b_{ij}+t_{ij}-1}$$

$$=\mathrm{beta}\left(F_{ij};a_{ij}+s_{ij},b_{ij}+t_{ij}\right)\blacksquare$$

(due to definition of beta density function specified in equation 4.12)

In brief, we have equation 4.1.13 for calculating posterior beta density function $\beta(F_{ij}|\mathcal{D})$.

$$\beta\left(F_{ij}\middle|\mathcal{D}\right)=\mathrm{beta}\left(F_{ij};a_{ij}+s_{ij},b_{ij}+t_{ij}\right)\tag{4.1.13}$$

Note that equation 4.1.13 is an extension of equation 4.1.3 in case of multi-node BN. Equation 4.1.13 is corollary 6.7 proved by similar way in (Neapolitan, 2003, p. 347) to which I referred. Applying equations 4.1.9 and 4.1.13, it is easy to calculate updated probability $P(X_i=1|PA_{ij}, \mathcal{D})$ of variable $X_i$ given its parent instance $PA_{ij}$ as follows:

$$P\left(X_i=1\middle|PA_{ij},\mathcal{D}\right)=E\left(F_{ij}\middle|\mathcal{D}\right)=\frac{a_{ij}+s_{ij}}{N_{ij}+M_{ij}}\tag{4.1.14}$$

Where $N_{ij}=a_{ij}+b_{ij}$ and $M_{ij}=s_{ij}+t_{ij}$. It is easy to recognize that equation 4.1.14 is an extension of equation 4.1.4 in case of multi-node BN. Hence, $F_{ij}$ is estimated by equation 4.1.14 under squared-error loss function with binomial sampling and prior beta distribution. In general, in case of binomial distribution, if we have the real/trust BN embedded in the expanded augmented network like figure 4.1.3 and each parameter node $F_{ij}$ has a prior beta distribution $\beta(F_{ij}; a_{ij}, b_{ij})$ and each hypothesis node $X_i$ has the prior conditional probability $P(X_i=1|PA_{ij}) = E(F_{ij}) = \frac{a_{ij}}{N_{ij}}$, the parameter learning process based on a set of evidences is to calculate posterior density function $\beta(F_{ij}|\mathcal{D})$ and updated conditional probability $P(X_i=1|PA_{ij},\mathcal{D})$. Indeed, we have $\beta(F_{ij}|\mathcal{D}) = beta(F_{ij}; a_{ij}+s_{ij}, b_{ij}+t_{ij})$ and $P(X_i=1|PA_{ij},\mathcal{D}) = E(F_{ij}|\mathcal{D}) = \frac{a_{ij}+s_{ij}}{N_{ij}+M_{ij}}$.

**Example 4.1.2.** For illustrating parameter learning based on beta density function, suppose we have a set of 5 evidences $\mathcal{D}=\{X^{(1)}, X^{(2)}, X^{(3)}, X^{(4)}, X^{(5)}\}$ owing to network in figure 4.1.3. Evidence sample (evidence matrix) $\mathcal{D}$ is shown in table 4.1.1 (Neapolitan, 2003, p. 358).

| | $X_1$ | $X_2$ |
|---|---|---|
| $\boldsymbol{X^{(1)}}$ | $X_1^{(1)} = 1$ | $X_2^{(1)} = 1$ |
| $\boldsymbol{X^{(2)}}$ | $X_1^{(2)} = 1$ | $X_2^{(2)} = 1$ |
| $\boldsymbol{X^{(3)}}$ | $X_1^{(3)} = 1$ | $X_2^{(3)} = 1$ |
| $\boldsymbol{X^{(4)}}$ | $X_1^{(4)} = 1$ | $X_2^{(4)} = 0$ |
| $\boldsymbol{X^{(5)}}$ | $X_1^{(5)} = 0$ | $X_2^{(5)} = 0$ |

**Table 4.1.1.** Evidence sample corresponding to 5 trials (sample of size 5)

In order to interpret evidence sample $\mathcal{D}$ in table 4.1.1, for instance, the first evidence (vector) $X^{(1)} = \begin{pmatrix} X_1^{(1)} = 1 \\ X_2^{(1)} = 1 \end{pmatrix}$ implies that variable $X_2$=1 given $X_1$=1 occurs in the first trial. We need to compute all posterior density functions $\beta(F_{11}|\mathcal{D})$, $\beta(F_{21}|\mathcal{D})$, $\beta(F_{22}|\mathcal{D})$ and all updated conditional probabilities $P(X_1=1|\mathcal{D})$, $P(X_2=1|X_1=1,\mathcal{D})$, $P(X_2=1|X_1=0,\mathcal{D})$ from prior density functions $\beta(F_{11}; 1,1)$, $\beta(F_{21}; 1,1)$, $\beta(F_{22}; 1,1)$. As usual, let counter $s_{ij}$ ($t_{ij}$) be the number of evidences among 5 trials such that variable $X_i$ = 1 and $PA_{ij}$ = 1 ($PA_{ij}$ = 0), the following table 4.1.2 shows counters $s_{ij}$, $t_{ij}$ (s) and posterior density functions calculated based on these counters; please see equation 4.1.13 for more details about updating posterior density functions. For instance, the number of rows (evidences) in table 4.1.1 such that $X_2$=1 given $X_1$=1 is 3, which causes $s_{21}$ = 3 in table 4.1.2.

| | |
|---|---|
| $s_{11}$=1+1+1+1+0=4 | $t_{11}$=0+0+0+0+1=1 |
| $s_{21}$=1+1+1+0+0=3 | $t_{21}$=0+0+0+0+1=1 |
| $s_{22}$=0+0+0+0+0=0 | $t_{21}$=0+0+0+0+1=1 |

$\beta(F_{11}|\mathcal{D}) = \beta(F_{11}; a_{11}+s_{11}, b_{11}+t_{11}) = \beta(F_{11}; 1+4, 1+1) = \beta(F_{11}; 5, 2)$

$\beta(F_{21}|\mathcal{D}) = \beta(F_{21}; a_{21}+s_{21}, b_{21}+t_{21}) = \beta(F_{21}; 1+3, 1+1) = \beta(F_{11}; 4, 2)$

$\beta(F_{22}|\mathcal{D}) = \beta(F_{22}; a_{22}+s_{22}, b_{22}+t_{22}) = \beta(F_{22}; 1+0, 1+1) = \beta(F_{11}; 1, 2)$

**Table 4.1.2.** Posterior density functions calculated based on count numbers $s_{ij}$ and $t_{ij}$

When posterior density functions are determined, it is easy to compute updated conditional probabilities $P(X_1=1|\mathcal{D})$, $P(X_2=1|X_1=1,\mathcal{D})$, and $P(X_2=1|X_1=0,\mathcal{D})$ as conditional expectations of $F_{11}$, $F_{21}$, and $F_{22}$, respectively according to equation 4.1.14. Table 4.1.3 expresses such updated conditional probabilities.

$$P(X_1 = 1|\mathcal{D}) = E(F_{11}|\mathcal{D}) = \frac{5}{5+2} = \frac{5}{7}$$

$$P(X_2 = 1|X_1 = 1, \mathcal{D}) = E(F_{21}|\mathcal{D}) = \frac{4}{4+2} = \frac{2}{3}$$
$$P(X_2 = 1|X_1 = 0, \mathcal{D}) = E(F_{22}|\mathcal{D}) = \frac{1}{1+2} = \frac{1}{3}$$

**Table 4.1.3.** Updated CPTs of $X_1$ and $X_2$

Note that inverted probabilities in CPTs such as $P(X_1=0|\mathcal{D})$, $P(X_2=0|X_1=1,\mathcal{D})$ and $P(X_2=0|X_1=0,\mathcal{D})$ are not mentioned because $X_i$ (s) are binary variables and so, $P(X_1=0|\mathcal{D}) = 1 – P(X_1=1|\mathcal{D}) = 2/7$, $P(X_2=0|X_1=1,\mathcal{D}) = 1 – P(X_2=1|X_1=1,\mathcal{D}) = 1/3$ and $P(X_2=0|X_1=0,\mathcal{D}) = 1 – P(X_2=1|X_1=0,\mathcal{D}) = 2/3$.

Now BN in figure 4.1.3 is updated based on evidence sample $\mathcal{D}$ and it is converted into the evolved BN with full of CPTs shown in figure 4.1.5■

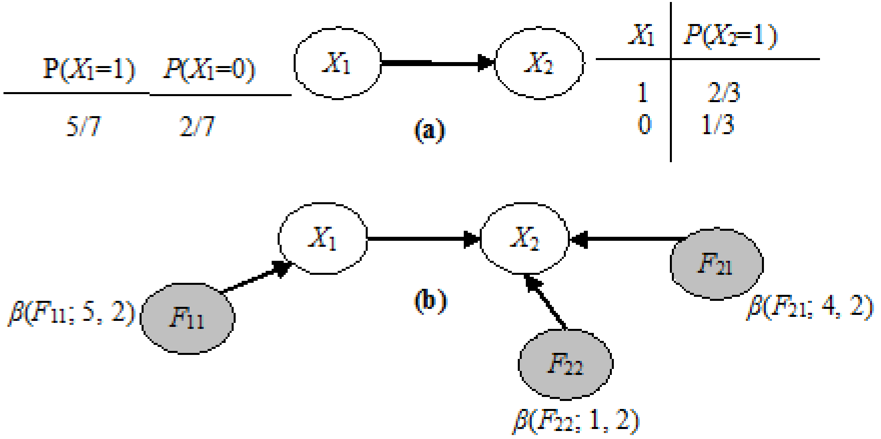


**Figure 4.1.5.** Updated version of BN (a) and binomial augmented BN (b)

It is easy to perform parameter learning by counting numbers $s_{ij}$ and $t_{ij}$ among sample according to expectation of beta density function as in equation 4.1.4 and 4.1.14 but a problem occurs when data in sample is missing. This problem is solved by expectation maximization (EM) algorithm mentioned in next sub-section 4.2.

The quality of parameter learning depends on how to specifies $a_{ij}$ and $b_{ij}$ in prior. We often set $a_{ij} = b_{ij}$ so that original probabilities $P(X_i) = 0.5$ and hence updated probabilities $P(X_i \mid \mathcal{D})$ are computed faithfully from sample. However, the number $N_{ij} = a_{ij} + b_{ij}$ also affects the quality of parameter learning. Hence, if a so-called equivalent sample size is satisfied, the quality of parameter learning is faithful. Another goal (Neapolitan, 2003, p. 351) of equivalent sample size is that updated parameters $a_{ij}$ and $b_{ij}$ based on sample will keep conditional independences entailed by the DAG.

According to definition 4.1.1 (Neapolitan, 2003, p. 351), suppose there is a binomial augmented BN and its parameters in full $\beta(F_{ij}; a_{ij}, b_{ij})$, for all $i$ and $j$, if there exists the number $N$ such that satisfying equation 4.1.15 then, the binomial augmented BN is called to have *equivalent sample size N*.

$$N_{ij} = a_{ij} + b_{ij} = P(PA_{ij}) * N \quad (\forall i, j) \qquad (4.1.15)$$

Where $P(PA_{ij})$ is probability of the $j^{th}$ parent instance of an $X_i$ and it is conventional that if $X_i$ has no parent then, $P(PA_{i1})=1$. The binomial augmented BN in figure 4.1.3 does not have prior equivalent sample size. If it is revised with $\beta(F_{11}; 2, 2)$, $\beta(F_{21}; 1,1)$, and $\beta(F_{22}; 1,1)$ then it has equivalent sample size 4 due to:

$4 = a_{11} + b_{11} = 1*4 = 4$ ($P(PA_{11})=1$ because $X_1$ has no parent)

$2 = a_{21} + b_{21} = P(X_1=1) *4 = ½*4 = 2$

$2 = a_{22} + b_{22} = P(X_1=0) *4 = ½*4 = 2$

If a binomial augmented BN has equivalent sample size $N$ then, for each node $X_i$, we have:

$$\sum_{j=1}^{q_i} N_{ij} = \sum_{j=1}^{q_i} P\big(PA_{ij}\big) * N = N \sum_{j=1}^{q_i} P\big(PA_{ij}\big) = N$$

Where $q_i$ is the number instances of parents of $X_i$. If $X_i$ has no parent then, $q_i$=1.

According to theorem 4.1.1 (Neapolitan, 2003, p. 353), suppose there is a binomial augmented BN and its parameters in full $\beta(F_{ij}; a_{ij}, b_{ij})$, for all $i$ and $j$, if there exists the number $N$ such that satisfying equation 4.1.16 then, the binomial augmented BN has equivalent sample size $N$ and the embedded BN has uniform joint probability distribution.

$$a_{ij} = b_{ij} = \frac{N}{2q_i} \tag{4.1.16}$$

Where $q_i$ is the number instances of parents of $X_i$. If $X_i$ has no parent then, $q_i$=1. It is easy to prove this theorem, we have:

$$\text{For all } i \text{ and } j, N_{ij} = a_{ij} + b_{ij} = \frac{2N}{2q_i} = \frac{1}{q_i} N = P\big(PA_{ij}\big) * N$$

According to theorem 4.1.2 (Neapolitan, 2003, p. 353), suppose there is a binomial augmented BN and its parameters in full $\beta(F_{ij}; a_{ij}, b_{ij})$, for all $i$ and $j$, if there exists the number $N$ such that satisfying equation 4.1.17 then, the binomial augmented BN has equivalent sample size $N$.

$$\begin{gathered} a_{ij} = P\big(X_i = 1\big|PA_{ij}\big) * P\big(PA_{ij}\big) * N \\ b_{ij} = P\big(X_i = 0\big|PA_{ij}\big) * P\big(PA_{ij}\big) * N \end{gathered} \tag{4.1.17}$$

Where $q_i$ is the number instances of parents of $X_i$. If $X_i$ has no parent then, $q_i$=1. It is easy to prove this theorem, we have:

$$\begin{aligned} \text{For all } i \text{ and } j, N_{ij} &= a_{ij} + b_{ij} \\ &= P\big(X_i = 1\big|PA_{ij}\big) * P\big(PA_{ij}\big) * N + P\big(X_i = 0\big|PA_{ij}\big) * P\big(PA_{ij}\big) * N \\ &= P\big(PA_{ij}\big) * N * \Big(P\big(X_i = 1\big|PA_{ij}\big) + P\big(X_i = 0\big|PA_{ij}\big)\Big) \\ &= P\big(PA_{ij}\big) * N * \Big(P\big(X_i = 1\big|PA_{ij}\big) + 1 - P\big(X_i = 1\big|PA_{ij}\big)\Big) = P\big(PA_{ij}\big) * N \blacksquare \end{aligned}$$

According to definition 4.1.2 (Neapolitan, 2003, p. 354), two binomial augmented BNs: ($G_1$, $F^{(G1)}$, $\rho^{(G1)}$) and ($G_2$, $F^{(G2)}$, $\rho^{(G2)}$) are called *equivalent* (or *augmented equivalent*) if they satisfy following conditions:

1. $G_1$ and $G_2$ are Markov equivalent.
2. The probability distributions in their embedded BNs ($G_1$, $P_1$) and ($G_2$, $P_2$) are the same, $P_1 = P_2$.
3. Of course, $\rho^{(G1)}$ and $\rho^{(G2)}$ are beta distributions, $\rho^{(G1)} = \beta^{(G2)}$ and $\rho^{(G2)} = \beta^{(G2)}$.
4. They share the same equivalent size.

Note that we can make some mapping so that a node $X_i$ in ($G_1$, $F^{(G1)}$, $\beta^{(G1)}$) is also node $X_i$ in ($G_2$, $F^{(G2)}$, $\beta^{(G2)}$) and a parameter $F_i$ in ($G_1$, $F^{(G1)}$, $\beta^{(G1)}$) is also parameter $F_i$ in ($G_2$, $F^{(G2)}$, $\beta^{(G2)}$) if ($G_1$, $F^{(G1)}$, $\beta^{(G1)}$) and ($G_2$, $F^{(G2)}$, $\beta^{(G2)}$) are equivalent.

Given binomial sample $\mathcal{D}$ and two binomial augmented BNs ($G_1$, $F^{(G1)}$, $\rho^{(G1)}$) and ($G_2$, $F^{(G2)}$, $\rho^{(G2)}$), according to lemma 4.1.1 (Neapolitan, 2003, p. 354), if such two augmented BNs are equivalent then, we have:

$$P_1(\mathcal{D}|G_1) = P_2(\mathcal{D}|G_1) \tag{4.1.18}$$

Where $P_1(\mathcal{D} \mid G_1)$ and $P_2(\mathcal{D} \mid G_2)$ are probabilities of sample $\mathcal{D}$ given parameters of $G_1$ and $G_2$, respectively. They are likelihood functions which are mentioned in equation 4.1.11.

$$P_1(\mathcal{D}|G_1) = P_1\left(\mathcal{D}\middle|F_1^{(G_1)}, F_2^{(G_1)}, \dots, F_n^{(G_1)}\right) = \prod_{i=1}^{n}\prod_{j=1}^{q_i}\left(F_{ij}^{(G_1)}\right)^{s_{ij}}\left(1 - F_{ij}^{(G_1)}\right)^{t_{ij}}$$

$$P_2(\mathcal{D}|G_2) = P_2\left(\mathcal{D}\middle|F_1^{(G_2)}, F_2^{(G_2)}, \dots, F_n^{(G_2)}\right) = \prod_{i=1}^{n}\prod_{j=1}^{q_i}\left(F_{ij}^{(G_2)}\right)^{s_{ij}}\left(1 - F_{ij}^{(G_2)}\right)^{t_{ij}}$$

Equation 4.1.18 specifies a so-called likelihood equivalence. In other words, if two augmented BNs are equivalent then, likelihood equivalence is obtained. Note, $F_{ij}^{(G_k)}$ denotes parameter $F_{ij}$ in BN ($G_k$, $P_k$).

According to corollary 4.1.1 (Neapolitan, 2003, p. 355), given binomial sample $\mathcal{D}$ and two binomial augmented BNs ($G_1$, $F^{(G1)}$, $\rho^{(G1)}$) and ($G_2$, $F^{(G2)}$, $\rho^{(G2)}$), if such two augmented BNs are equivalent then, two updated probabilities corresponding two embedded BNs ($G_1$, $P_1$) and ($G_2$, $P_2$) are equal as follows:

$$P_1\left(X_i^{(G_1)} = 1\middle|PA_{ij}^{(G_1)}, \mathcal{D}\right) = P_2\left(X_i^{(G_2)} = 1\middle|PA_{ij}^{(G_2)}, \mathcal{D}\right) \tag{4.1.19}$$

These update probabilities are specified by equation 4.1.14.

$$P_1\left(X_i^{(G_1)} = 1\middle|PA_{ij}^{(G_1)}, \mathcal{D}\right) = E\left(F_{ij}^{(G_1)}\middle|\mathcal{D}\right) = \frac{a_{ij}^{(G_1)} + s_{ij}^{(G_1)}}{N_{ij}^{(G_1)} + M_{ij}^{(G_1)}}$$

$$P_2\left(X_i^{(G_2)} = 1\middle|PA_{ij}^{(G_2)}, \mathcal{D}\right) = E\left(F_{ij}^{(G_2)}\middle|\mathcal{D}\right) = \frac{a_{ij}^{(G_2)} + s_{ij}^{(G_2)}}{N_{ij}^{(G_2)} + M_{ij}^{(G_2)}}$$

Note, $X_i^{(G_k)}$ denotes node $X_i$ in $G_k$ and hence, other notations are similar.

Because this report focuses on discrete BN, parameter $F$ in augmented BN is assumed to conform beta distribution, which derives beautiful results in calculating updated probability. We should skim some other results related the fact that $F$ follows some distribution so that the density function $\rho$ in augmented BN ($G$, $F^{(G)}$, $\rho^{(G)}$) is arbitrary. Equation 4.1.5 is still kept.

$$P\left(X_i = 1\middle|PA_{ij}, F_{i1}, F_{i2}, \dots, F_{ij}, \dots, F_{iq_i}\right) = P\left(X_i = 1\middle|PA_{ij}, F_{ij}\right) = F_{ij}$$

Global and local parameter independences (please see equations 4.1.7 and 4.1.8) are kept intact as follows:

$$\begin{gathered}\rho(F_i) = \prod_{j=1}^{q_i}\rho\left(F_{ij}\right)\\ \rho(F_1, F_2, \dots, F_i, \dots, F_n) = \prod_{i=1}^{n}\prod_{j=1}^{q_i}\rho\left(F_{ij}\right)\end{gathered} \tag{4.1.20}$$

From global and local parameter independences, $\rho(F_1, F_2, \dots, F_n)$ is defined based on many $\rho(F_i)$ which in turn is defined based on many $\rho(F_{ij})$.

Probability $P(X_i=1 \mid PA_{ij})$ is still expectation of $F_{ij}$ (Neapolitan, 2003, p. 334) given prior density function $\rho(F_{ij})$ with recall that $0 \le F_{ij} \le 1$.

$$P\left(X_i = 1\middle|PA_{ij}\right) = E\left(F_{ij}\right) = \int_{F_{ij}} F_{ij}\rho\left(F_{ij}\right)\mathrm{d}F_{ij} \tag{4.1.21}$$

Equation 4.1.21 is not as specific as equation 4.1.9 because $\rho$ is arbitrary; please see the proof of equation 4.1.9 to know how to prove equation 4.1.21. Based on binomial trials and mutual independence, the probability of evidences corresponding to variable $X_i$ over $m$ trials is:

$$P\left(X_i^{(1)}, X_i^{(2)}, \dots, X_i^{(M)} \middle| PA_i, F_i\right) = \prod_{u=1}^{m} P\left(X_i^{(u)} \middle| PA_i, F_i\right) \tag{4.1.22}$$

Equation 4.1.22 is not as specific as equation 4.1.10 because $\rho$ is arbitrary. Likelihood function $P(\mathcal{D}|F_1, F_2, \dots, F_n)$ is specified by equation 4.1.23.

$$\begin{aligned} P(\mathcal{D}|F_1, F_2, \dots, F_n) &= P\left(X^{(1)}, X^{(2)}, \dots, X^{(m)} \middle| F_1, F_2, \dots, F_n\right) \\ &= \prod_{i=1}^{n} \prod_{u=1}^{m} P\left(X_i^{(u)} \middle| PA_i, F_i\right) \end{aligned} \tag{4.1.23}$$

Equation 4.1.23 is not as specific as equation 4.1.11 because $\rho$ is arbitrary; please see the proof of equation 4.1.11 to know how to prove equation 4.1.23. Likelihood function $P(\mathcal{D}|F_i)$ with regard to only parameter $F_i$ is specified by equation 4.1.24.

$$\begin{aligned} P(\mathcal{D}|F_i) &= P\left(X^{(1)}, X^{(2)}, \dots, X^{(m)} \middle| F_i\right) \\ &= \left(\prod_{u=1}^{m} P\left(X_i^{(u)} \middle| PA_i, F_i\right)\right) \\ &\quad * \left(\prod_{j=1, j \neq i}^{n} \int_{F_j} \prod_{u=1}^{m} P\left(X_j^{(u)} \middle| PA_j, F_j\right) \rho(F_j) \mathrm{d}F_j\right) \end{aligned} \tag{4.1.24}$$

Following is the proof of equation 4.1.24 (Neapolitan, 2003, p. 339).

$$P(\mathcal{D}|F_i) = \int_{F_j \neq F_i} P(\mathcal{D}|F_1, F_2, \dots, F_n) \prod_{j \neq i} \rho(F_j) \mathrm{d}F_j$$

(Due to law of total probability)

$$= \int_{F_j \neq F_i} P\left(X^{(1)}, X^{(2)}, \dots, X^{(m)} \middle| F_1, F_2, \dots, F_n\right) \prod_{j \neq i} \rho(F_j) \mathrm{d}F_j$$

(Because evidences are mutually independent)

$$= \int_{F_j \neq F_i} \prod_{j=1}^{n} \prod_{u=1}^{m} P\left(X_j^{(u)} \middle| PA_j, F_j\right) \prod_{j \neq i} \rho(F_j) \mathrm{d}F_j$$

(Due to equation 4.1.23)

$$= \left(\prod_{u=1}^{m} P\left(X_i^{(u)} \middle| PA_i, F_i\right)\right) * \left(\int_{F_j \neq F_i} \prod_{j=1, j \neq i}^{n} \prod_{u=1}^{m} P\left(X_j^{(u)} \middle| PA_j, F_j\right) \prod_{j \neq i} \rho(F_j) \mathrm{d}F_j\right)$$

$$= \left(\prod_{u=1}^{m} P\left(X_i^{(u)} \middle| PA_i, F_i\right)\right) * \left(\prod_{j=1, j \neq i}^{n} \int_{F_j} \prod_{u=1}^{m} P\left(X_j^{(u)} \middle| PA_j, F_j\right) \rho(F_j) \mathrm{d}F_j\right)$$

Marginal probability $P(\mathcal{D})$ of evidence sample $\mathcal{D}$ is:

$$P(\mathcal{D}) = P\left(X^{(1)}, X^{(2)}, \dots, X^{(m)}\right) = \prod_{i=1}^{n} \prod_{u=1}^{m} \int_{F_i} P\left(X_i^{(u)} \middle| PA_i, F_i\right) \rho(F_i) \mathrm{d}F_i \tag{4.1.25}$$

Equation 4.1.25 is not as specific as equation 4.1.12 because $\rho$ is arbitrary; please see the proof of equation 4.1.12 to know how to prove equation 4.1.25. Equation 4.1.26 specifies posterior density function $\rho(F_i \mid \mathcal{D})$ with support of equations 4.1.24 and 4.1.25.

$$\rho(F_i|\mathcal{D}) = \frac{P(\mathcal{D}|F_i)\rho(F_i)}{P(\mathcal{D})} \tag{4.1.26}$$

Posterior density function $\rho(F_{ij} \mid \mathcal{D})$ is determined based on posterior density function $\rho(F_i \mid \mathcal{D})$ as follows:

$$\rho\left(F_{ij}\middle|\mathcal{D}\right) = \int\limits_{\substack{F_{ik}\\ k\neq j}} \rho(F_i|\mathcal{D}) \prod_{k=1}^{q_i} \mathrm{d}F_{ik} = \int\limits_{\substack{F_{ik}\\ k\neq j}} \rho\left(F_{i1}, F_{i2}, \dots, F_{ij}, \dots, F_{iq_i}\middle|\mathcal{D}\right) \prod_{k=1}^{q_i} \mathrm{d}F_{ik} \tag{4.1.27}$$

Therefore, updated probability $P(X_i=1 \mid PA_{ij}, \mathcal{D})$ is expectation of $F_{ij}$ given posterior density function $\rho(F_{ij} \mid \mathcal{D})$.

$$P\left(X_i = 1\middle|PA_{ij}, \mathcal{D}\right) = E\left(F_{ij}\middle|\mathcal{D}\right) = \int\limits_{F_{ij}} F_{ij}\rho\left(F_{ij}\middle|\mathcal{D}\right)\mathrm{d}F_{ij} \tag{4.1.28}$$

Note, equation 4.1.28 is like equation 4.1.21 except that prior density function $\rho(F_{ij})$ is replaced by posterior density function $\rho(F_{ij} \mid \mathcal{D})$.

## 4.2. Parameter learning with binomial incomplete data

In practice there are some evidences in $\mathcal{D}$ such as $X^{(u)}$ (s) which lack information and thus, it stimulates the question "How to update network from missing data". We must address this problem by artificial intelligence techniques, namely, Expectation Maximization (EM) algorithm – a famous technique solving estimation of missing data. EM algorithm has two steps such as Expectation step (E-step) and Maximization step (M-step), which aims to improve parameters after a number of iterations; please read (Borman, 2004) for more details about EM algorithm. We will know thoroughly these steps by reviewing above example shown in table 4.1.1, in which there is the set of 5 evidences $\mathcal{D}=\{X^{(1)}, X^{(2)}, X^{(3)}, X^{(4)}, X^{(5)}\}$ along with network in figure 4.1.3 but the evidences $X^{(2)}$ and $X^{(5)}$ have not data yet. Table 4.2.1 shows such missing data (Neapolitan, 2003, p. 359).

| | $X_1$ | $X_2$ |
|---|---|---|
| $\boldsymbol{X^{(1)}}$ | $X_1^{(1)} = 1$ | $X_2^{(1)} = 1$ |
| $\boldsymbol{X^{(2)}}$ | $X_1^{(2)} = 1$ | $X_2^{(2)} =$ $\boldsymbol{v_1}$**?** |
| $\boldsymbol{X^{(3)}}$ | $X_1^{(3)} = 1$ | $X_2^{(3)} = 1$ |
| $\boldsymbol{X^{(4)}}$ | $X_1^{(4)} = 1$ | $X_2^{(4)} = 0$ |
| $\boldsymbol{X^{(5)}}$ | $X_1^{(5)} = 0$ | $X_2^{(5)} =$ $\boldsymbol{v_2}$**?** |

**Table 4.2.1.** Evidence sample with missing data

**Example 4.2.1.** As known, count numbers $s_{21}$, $t_{21}$ and $s_{22}$, $t_{22}$ can't be computed directly, it means that it is not able to compute directly posterior density functions $\beta(F_{11}|\mathcal{D})$, $\beta(F_{21}|\mathcal{D})$, and $\beta(F_{22}|\mathcal{D})$. It is necessary to determine missing values $v_1$ and $v_2$. Because $v_1$ and $v_2$ are binary values (1 and 0), we calculate their occurrences. So, evidence $X^{(2)}$ is split into two $X$ ‘$^{(2)}$ (s) corresponding to two values 1 and 0 of $v_1$. Similarly, evidence $X^{(5)}$ is split into two $X$ ‘$^{(5)}$ (s) corresponding to two values 1 and 0 of $v_2$. Table 4.2.2 shows new split evidences for missing data.

| | $X_1$ | $X_2$ | #Occurrences |
|---|---|---|---|
| $\boldsymbol{X^{(1)}}$ | $X_1^{(1)} = 1$ | $X_2^{(1)} = 1$ | 1 |
| $\boldsymbol{X^{`(2)}}$ | $X_1{}^{’(2)} = 1$ | $X_2{}^{’(2)} = 1$ | $\#n_{11}$ |
| $\boldsymbol{X^{`(2)}}$ | $X_1{}^{’(2)} = 1$ | $X_2{}^{’(2)} = 0$ | $\#n_{10}$ |
| $\boldsymbol{X^{(3)}}$ | $X_1^{(3)} = 1$ | $X_2^{(3)} = 1$ | 1 |
| $\boldsymbol{X^{(4)}}$ | $X_1^{(4)} = 1$ | $X_2^{(4)} = 0$ | 1 |

| | | | |
|---|---|---|---|
| $\boldsymbol{X}'^{(5)}$ | $X_1'^{(5)} = 0$ | $X_2'^{(5)} = 1$ | $\#n_{21}$ |
| $\boldsymbol{X}'^{(5)}$ | $X_1'^{(5)} = 0$ | $X_2'^{(5)} = 0$ | $\#n_{20}$ |

**Table 4.2.2.** New split evidences for missing data

The number $\#n_{11}$ ($\#n_{10}$) of occurrences of $v_1$=1 ($v_1$=0) is estimated by the probability of $X_2 = 1$ given $X_1 = 1$ ($X_2 = 0$ given $X_1 = 1$) with assumption that $a_{21} = 1$ and $b_{21} = 1$ as in figure 4.1.3.

$$\#n_{11} = P(X_2 = 1|X_1 = 1) = E(F_{21}) = \frac{a_{21}}{a_{21} + b_{21}} = \frac{1}{2}$$

$$\#n_{10} = P(X_2 = 0|X_1 = 1) = 1 - P(X_2 = 1|X_1 = 1) = 1 - \frac{1}{2} = \frac{1}{2}$$

Similarly, the number $\#n_{21}$ ($\#n_{20}$) of occurrences of $v_2$=1 ($v_2$=0) is estimated by the probability of $X_2 = 1$ given $X_1 = 0$ ($X_2 = 0$ given $X_1 = 0$) with assumption that $a_{22} = 1$ and $b_{22} = 1$ as in figure 4.1.3.

$$\#n_{21} = P(X_2 = 1|X_1 = 0) = E(F_{22}) = \frac{a_{22}}{a_{22} + b_{22}} = \frac{1}{2}$$

$$\#n_{20} = P(X_2 = 0|X_1 = 0) = 1 - P(X_2 = 1|X_1 = 0) = 1 - \frac{1}{2} = \frac{1}{2}$$

When $\#n_{11}$, $\#n_{10}$, $\#n_{21}$, and $\#n_{20}$ are determined, missing data is filled fully and evidence sample $\mathcal{D}$ is completed as in table 4.2.3.

| | $X_1$ | $X_2$ | #Occurrences |
|---|---|---|---|
| $\boldsymbol{X}^{(1)}$ | $X_1^{(1)} = 1$ | $X_2^{(1)} = 1$ | 1 |
| $\boldsymbol{X}'^{(2)}$ | $X_1'^{(2)} = 1$ | $X_2'^{(2)} = 1$ | 1/2 |
| $\boldsymbol{X}'^{(2)}$ | $X_1'^{(2)} = 1$ | $X_2'^{(2)} = 0$ | 1/2 |
| $\boldsymbol{X}^{(3)}$ | $X_1^{(3)} = 1$ | $X_2^{(3)} = 1$ | 1 |
| $\boldsymbol{X}^{(4)}$ | $X_1^{(4)} = 1$ | $X_2^{(4)} = 0$ | 1 |
| $\boldsymbol{X}'^{(5)}$ | $X_1'^{(5)} = 0$ | $X_2'^{(5)} = 1$ | 1/2 |
| $\boldsymbol{X}'^{(5)}$ | $X_1'^{(5)} = 0$ | $X_2'^{(5)} = 0$ | 1/2 |

**Table 4.2.3.** Complete evidence sample in E-step of EM algorithm

In general, the essence of this task – estimating missing values by *expectations* of $F_{21}$ and $F_{22}$ based on previous parameters $a_{21}$, $b_{21}$, $a_{22}$, and $b_{22}$ of beta density functions is E-step in EM algorithm. Of course, in E-step, when missing values are estimated, it is easy to determine counters $s_{11}$, $t_{11}$, $s_{21}$, $t_{21}$, $s_{22}$, and $t_{22}$. Recall that counters $s_{11}$ and $t_{11}$ are numbers of evidences such that $X_1 = 1$ and $X_1 = 0$, respectively. Counters $s_{21}$ and $t_{21}$ ($s_{22}$ and $t_{22}$) are numbers of evidences such that $X_2 = 1$ and $X_2 = 0$ given $X_1 = 1$ ($X_2 = 1$ and $X_2 = 0$ given $X_1 = 0$), respectively. In fact, these counters are ultimate results of E-step. From complete sample $\mathcal{D}$ in table 4.2.3, we have table 4.2.4 showing such ultimate results of E-step:

| | |
|---|---|
| $s_{11} = 1 + \frac{1}{2} + \frac{1}{2} + 1 + 1 = 4$ | $t_{11} = \frac{1}{2} + \frac{1}{2} = 1$ |
| $s_{21} = 1 + \frac{1}{2} + 1 = \frac{5}{2}$ | $t_{21} = \frac{1}{2} + 1 = \frac{3}{2}$ |
| $s_{22} = \frac{1}{2}$ | $t_{22} = \frac{1}{2}$ |

**Table 4.2.4.** Counters $s_{11}$, $t_{11}$, $s_{21}$, $t_{21}$, $s_{22}$, and $t_{22}$ from estimated values (of missing values)

The next step of EM algorithm, M-step is responsible for updating posterior density functions $\beta(F_{11}|\mathcal{D})$, $\beta(F_{21}|\mathcal{D})$, and $\beta(F_{22}|\mathcal{D})$, which leads to calculate updated probabilities $P(X_1=1|\mathcal{D})$, $P(X_2=1|X_1=1,\mathcal{D})$, and $P(X_2=1|X_1=0,\mathcal{D})$, based on current counters $s_{11}$, $t_{11}$, $s_{21}$, $t_{21}$, $s_{22}$, and $t_{22}$ from complete evidence sample $\mathcal{D}$ (table 4.2.3). Table 4.2.5 shows results of M-step which are posterior density functions $\beta(F_{11}|\mathcal{D})$, $\beta(F_{21}|\mathcal{D})$, and $\beta(F_{22}|\mathcal{D})$ along with updated probabilities (updated CPT) such as $P(X_1=1|\mathcal{D})$, $P(X_2=1|X_1=1,\mathcal{D})$, and $P(X_2=1|X_1=0,\mathcal{D})$.

$$\beta(F_{11}|\mathcal{D}) = \beta(F_{11}; a_{11}+s_{11}, b_{11}+t_{11}) = \beta(F_{11}; 1+4, 1+1) = \beta(F_{11}; 5,2)$$

$$\beta(F_{21}|\mathcal{D}) = \beta(F_{21}; a_{21}+s_{21}, b_{21}+t_{21}) = \beta\left(F_{21}; 1+\frac{5}{2}, 1+\frac{3}{2}\right) = \beta\left(F_{21}; \frac{7}{2}, \frac{5}{2}\right)$$

$$\beta(F_{22}|\mathcal{D}) = \beta(F_{22}; a_{22}+s_{22}, b_{22}+t_{22}) = \beta\left(F_{21}; 1+\frac{1}{2}, 1+\frac{1}{2}\right) = \beta\left(F_{22}; \frac{3}{2}, \frac{3}{2}\right)$$

$$P(X_1=1|\mathcal{D}) = E(F_{11}|\mathcal{D}) = \frac{5}{5+2} = \frac{5}{7}$$

$$P(X_2=1|X_1=1,\mathcal{D}) = E(F_{21}|\mathcal{D}) = \frac{7/2}{7/2+5/2} = \frac{7}{12}$$

$$P(X_2=1|X_1=0,\mathcal{D}) = E(F_{22}|\mathcal{D}) = \frac{3/2}{3/2+3/2} = \frac{1}{2}$$

**Table 4.2.5.** Posterior density functions and updated probabilities in M-step of EM algorithm

Note that origin parameters such as $a_{11}=1$, $b_{11}=1$, $a_{21}=1$, $b_{21}=1$, $a_{22}=1$, and $b_{22}=1$ (see figure 4.1.3) are kept intact in the task of updating posterior density functions $\beta(F_{11}|\mathcal{D})$, $\beta(F_{21}|\mathcal{D})$, and $\beta(F_{22}|\mathcal{D})$. For example, $\beta(F_{11}|\mathcal{D}) = \beta(F_{11}; a_{11}+s_{11}, b_{11}+t_{11}) = \beta(F_{11}; 1+4, 1+1) = \beta(F_{11}; 5,2)$. After the updating task, these parameters are changed into new values; concretely, $a_{11}=5$, $b_{11}=2$, $a_{21}=7/2$, $b_{21}=5/2$, $a_{22}=3/2$, and $b_{22}=3/2$. These parameters updated with new values, which are called as updated parameters, are in turn used for the new iteration of EM algorithm■

The process of such two steps (E-step and M-step) repeated more and more brings out the EM algorithm. In general, EM algorithm is the iterative algorithm having many iterations and each iteration has two steps: E-step and M-step. Given the $k^{th}$ iteration in EM algorithm whose two steps such as E-step and M-step are summarized as follows:

1. *E-step*. Missing values are estimated based on expectations of $F_{ij}$ with regard to previous ($(k–1)^{th}$) parameters $a_{ij}$ and $b_{ij}$. Current ($k^{th}$) counters $s_{ij}$ and $t_{ij}$ are calculated with estimated values (of such missing values). Table 4.2.4 shows such current counters which are ultimate results of E-step.
2. *M-step*. Posterior density functions and updated probabilities (CPT) are calculated based on current ($k^{th}$) counters $s_{ij}$ and $t_{ij}$. Of course, $a_{ij}$ and $b_{ij}$ are updated because they are parameters of (beta) density functions. Table 4.2.5 shows results of M-step. Terminating algorithm if stop condition becomes true, otherwise, reiterating step 1. The stop condition may be "posterior density functions and updated probabilities are not changed significantly", "the number of iterations approaches *k times*" or "there is no missing value".

After $k^{th}$ iteration, the limit

$$\lim_{k\to+\infty} E\left(F_{ij}\middle|\mathcal{D}\right)^{(k)} = \lim_{k\to+\infty} \frac{a_{ij}^{(k)} + s_{ij}^{(k)}}{a_{ij}^{(k)} + s_{ij}^{(k)} + b_{ij}^{(k)} + t_{ij}^{(k)}}$$

will approach a certain limit. Note, the upper script ($k$) denotes the $k^{th}$ iteration. Don't worry about the case of infinite iterations, we will obtain optimal probability $P(X_i=1|PA_{ij}, \mathcal{D}) = \lim_{k\to+\infty} E\left(F_{ij}\middle|\mathcal{D}\right)^{(k)}$ if $k$ is large enough. This limit is noted similarly as equation 6.17 in (Neapolitan, 2003, p. 361). EM algorithm for learning parameters in BN is also mentioned particularly in (Neapolitan, 2003, pp. 359-363).

**Example 4.2.2.** Go backing the example of missing data, the results of EM algorithm at the first iteration are summarized from table 4.2.5, as follows:

$$a_{11} = 5, b_{11} = 2, a_{21} = \frac{7}{2}, b_{21} = \frac{5}{2}, a_{22} = \frac{3}{2}, b_{22} = \frac{3}{2}$$

$$P(X_1 = 1) = \frac{5}{7} \approx 0.71, P(X_2 = 1|X_1 = 1) = \frac{7}{12} \approx 0.58, P(X_2 = 1|X_1 = 0) = \frac{1}{2} = 0.5$$

When compared with the origin probabilities

$$P(X_1 = 1) = \frac{1}{2} = 0.5, P(X_2 = 1|X_1 = 1) = \frac{1}{2} = 0.5, P(X_2 = 1|X_1 = 0) = \frac{1}{2} = 0.5$$

There is significant change in these probabilities if the maximum deviation is pre-defined 0.05. It is easy for us to verify this assertion, concretely, |0.71 – 0.5| = 0.21 > 0.05. So it is necessary to run the EM algorithm at the second iteration.

At the second iteration, the E-step starts calculating the number #$n_{11}$ (#$n_{10}$) of occurrences of $v_1$=1 ($v_1$=0) and the number #$n_{21}$ (#$n_{20}$) of occurrences of $v_2$=1 ($v_2$=0) again:

$$\#n_{11} = P(X_2 = 1|X_1 = 1) = E(F_{21}) = \frac{a_{21}}{a_{21} + b_{21}} = \frac{7/2}{7/2 + 5/2} = \frac{7}{12}$$

$$\#n_{10} = P(X_2 = 0|X_1 = 1) = 1 - P(X_2 = 1|X_1 = 1) = 1 - \frac{7}{12} = \frac{5}{12}$$

$$\#n_{21} = P(X_2 = 1|X_1 = 0) = E(F_{22}) = \frac{a_{22}}{a_{22} + b_{22}} = \frac{3/2}{3/2 + 3/2} = \frac{1}{2}$$

$$\#n_{20} = P(X_2 = 0|X_1 = 0) = 1 - P(X_2 = 1|X_1 = 0) = 1 - \frac{1}{2} = \frac{1}{2}$$

When #$n_{11}$, #$n_{10}$, #$n_{21}$, and #$n_{20}$ are determined, missing data is filled fully and evidence sample $\mathcal{D}$ is completed as follows:

| | $X_1$ | $X_2$ | #Occurrences |
|---|---|---|---|
| $\boldsymbol{X}^{(1)}$ | $X_1^{(1)} = 1$ | $X_2^{(1)} = 1$ | 1 |
| $\boldsymbol{X}^{‘(2)}$ | $X_1’^{(2)} = 1$ | $X_2’^{(2)} = 1$ | 7/12 |
| $\boldsymbol{X}^{‘(2)}$ | $X_1’^{(2)} = 1$ | $X_2’^{(2)} = 0$ | 5/12 |
| $\boldsymbol{X}^{(3)}$ | $X_1^{(3)} = 1$ | $X_2^{(3)} = 1$ | 1 |
| $\boldsymbol{X}^{(4)}$ | $X_1^{(4)} = 1$ | $X_2^{(4)} = 0$ | 1 |
| $\boldsymbol{X}^{‘(5)}$ | $X_1’^{(5)} = 0$ | $X_2’^{(5)} = 1$ | 1/2 |
| $\boldsymbol{X}^{‘(5)}$ | $X_1’^{(5)} = 0$ | $X_2’^{(5)} = 0$ | 1/2 |

Recall that counters $s_{11}$ and $t_{11}$ are numbers of evidences such that $X_1$ = 1 and $X_1$ = 0, respectively. Counters $s_{21}$ and $t_{21}$ ($s_{22}$ and $t_{22}$) are numbers of evidences such that $X_2$ = 1 and $X_2$ = 0 given $X_1$ = 1 ($X_2$ = 1 and $X_2$ = 0 given $X_1$ = 0), respectively. These counters which are ultimate results of E-step are calculated as follows:

$$\boxed{\begin{array}{ll} s_{11} = 1 + \frac{7}{12} + \frac{5}{12} + 1 + 1 = 4 & t_{11} = \frac{1}{2} + \frac{1}{2} = 1 \\ s_{21} = 1 + \frac{7}{12} + 1 = \frac{31}{12} & t_{21} = \frac{5}{12} + 1 = \frac{17}{12} \\ s_{22} = \frac{1}{2} & t_{22} = \frac{1}{2} \end{array}}$$

Posterior density functions $\beta(F_{11}|\mathcal{D})$, $\beta(F_{21}|\mathcal{D})$, and $\beta(F_{22}|\mathcal{D})$, updated probabilities $P(X_1=1|\mathcal{D})$, $P(X_2=1|X_1=1,\mathcal{D})$, and $P(X_2=1|X_1=0,\mathcal{D})$ are updated at M-step as follows:

$$\beta(F_{11}|\mathcal{D}) = \beta(F_{11}; a_{11} + s_{11}, b_{11} + t_{11}) = \beta(F_{11}; 5 + 4, 2 + 1) = \beta(F_{11}; 9,3)$$

$$\beta(F_{21}|\mathcal{D}) = \beta(F_{21}; a_{21} + s_{21}, b_{21} + t_{21}) = \beta\left(F_{21}; \frac{7}{2} + \frac{31}{12}, \frac{5}{2} + \frac{17}{12}\right) = \beta\left(F_{21}; \frac{73}{12}, \frac{47}{12}\right)$$

$$\beta(F_{22}|\mathcal{D}) = \beta(F_{22}; a_{22} + s_{22}, b_{22} + t_{22}) = \beta\left(F_{21}; \frac{3}{2} + \frac{1}{2}, \frac{3}{2} + \frac{1}{2}\right) = \beta(F_{22}; 2{,}2)$$

$$P(X_1 = 1|\mathcal{D}) = E(F_{11}|\mathcal{D}) = \frac{9}{9+3} = \frac{3}{4} = 0.75$$

$$P(X_2 = 1|X_1 = 1, \mathcal{D}) = E(F_{21}|\mathcal{D}) = \frac{73/12}{73/12 + 47/12} = \frac{73}{120} \approx 0.61$$

$$P(X_2 = 1|X_1 = 0, \mathcal{D}) = E(F_{22}|\mathcal{D}) = \frac{2}{2+2} = \frac{1}{2} = 0.5$$

When compared with the previous probabilities

$$P(X_1 = 1) = \frac{5}{7} \approx 0.71, P(X_2 = 1|X_1 = 1) = \frac{7}{12} \approx 0.58, P(X_2 = 1|X_1 = 0) = \frac{1}{2} = 0.5$$

There is no significant change in these probabilities if the maximum deviation is pre-defined 0.05. It is easy for us to verify this assertion, concretely, |0.75 – 0.71| = 0.04 < 0.05, |0.61 – 0.58| = 0.03 < 0.05, and |0.5 – 0.5| = 0 < 0.05. So the EM algorithm is stopped with note that we can execute more iterations for EM algorithm in order to receive more precise results that updated probabilities are stable $\left(\lim_{k\to+\infty} E(F_{ij}|\mathcal{D})^{(k)} \text{ approaches certain limit}\right)$. Consequently, the Bayesian network in figure 4.1.3 is converted into the evolutional version specified in figure 4.2.1■

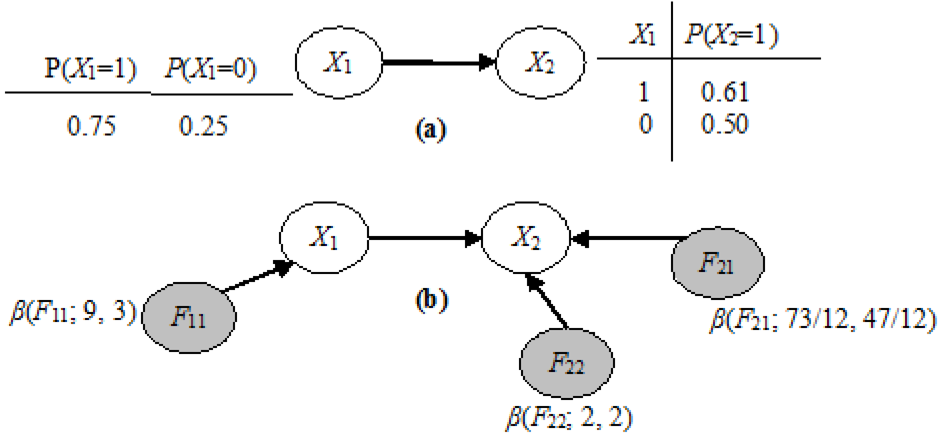


**Figure 4.2.1.** Updated version of BN (a) and binomial augmented BN (b) in case of missing data

In general, parameter learning is described thoroughly in this section. The next section mentions structure learning.

## 4.3. Parameter learning with multinomial complete data

Now each node *X* in BN is multinomial random variable whose possible values are 1, 2,…, *r*. Node *X* here is general case of discrete variable. As usual, $F = \Theta = (f_1, f_2, \ldots, f_r)$ is augmented variable associated with *X*, in which $f_k$ is parameter corresponding to *X*=*k*. Let *F* conform Dirichlet distribution as follows:

$$P(F) = \text{Dir}(F|a_1, a_2, \ldots, a_r) = \beta(F|a_1, a_2, \ldots, a_r) = \frac{\Gamma(N)}{\prod_{k=1}^{r} a_k} \prod_{k=1}^{r} (\theta_k)^{a_k - 1} \qquad (4.3.1)$$

Where,

$$N = \sum_{k=1}^{r} a_k$$
$$a_k > 0$$

Equation 4.3.1 is replication of equation 4.22, which is Dirichlet density function. The augmented BN is still as a triple ($G$, $F^{(G)}$, Dir$^{(G)}$) whereas the BN is denoted as a pair ($G$, $P$) which is still called embedded BN of ($G$, $F^{(G)}$, Dir$^{(G)}$). The probability $P(X = k)$ which is parameter of BN is prior probability as follows:

$$P(X=k)=E(f_k)=\frac{a_k}{N} \tag{4.3.2}$$

Note, $P(X=k)$ is CPT of $X$. Equation 4.3.2 is replication of equation 4.24. We also denote the vector of all evidences as $\mathcal{D} = (X^{(1)}, X^{(2)},\ldots, X^{(m)})$ which is also called the sample of size $m$. Suppose $\mathcal{D}$ is multinomial sample, we need to compute posterior density function Dir($F|\mathcal{D}$) and updated probability $P(X=k|\mathcal{D})$. Following equations 4.26 and 4.27, we have:

$$\mathrm{Dir}(F|\mathcal{D})=\mathrm{Dir}(F|a_1+s_1,a_2+s_2,\dots,a_r+s_r) \tag{4.3.3}$$

$$P(X=k|\mathcal{D})=E(f_k|\mathcal{D})=\frac{a_k+s_k}{N+M} \tag{4.3.4}$$

From equation 4.3.4, $P(X=k|\ \mathcal{D})$ representing updated CPT of $X$ is an estimate of $f_k$ under squared-error loss function. Equation 4.3.4 is corollary 7.1 in (Neapolitan, 2003, p. 383). Please pay attention to equations 4.3.3 and 4.3.4 because they are used to calculate posterior density function Dir($F|\mathcal{D}$) and updated probability (updated CPT) $P(X=k|\mathcal{D})$ of BN having one multinomial node.

Now we expand (augmented) BN with more than one hypothesis node. Suppose each $X_i$ has $r_i$ possible values (1, 2,…, $r_i$). If $X_i$ has a set of $p_i$ parent nodes and each parent node $X_k$ has $r_k$ possible values (1, 2,…, $r_k$), we add a set of $q_i=\sum_{k=1}^{p_i}r_k$ parameter variables $\{F_{i1}, F_{i2},\ldots, F_{iq_i}\}$ which, in turn, correspond to instances of parent nodes of $X_i$, namely $\{PA_{i1}, PA_{i2}, PA_{i3},\ldots, PA_{iq_i}\}$ where each $PA_{ij}$ is an instance of a parent node of $X_i$. For convenience, each $PA_{ij}$ is called a parent instance of $X_i$ and we let $PA_i=\{PA_{i1}, PA_{i2}, PA_{i3},\ldots, PA_{iq_i}\}$ be the vector or collection of parent instances of $X_i$. We also let $F_i=\{F_{i1}, F_{i2},\ldots, F_{iq_i}\}$ be the respective vector or collection of augmented variables $F_{i1}$ (s) attached to $X_i$. It is conventional that each $X_i$ has $q_i$ parent instances ($q_i \geq 0$); in other words, $q_i$ denotes the size of $PA_i$ and the size of $F_i$. We have equation 4.3.5 for connecting CPT of variable $X_i$ with Dirichlet density function of augmented variable $F_i$.

$$\begin{gathered}P\left(X_i=k\middle|PA_{ij},F_{i1},F_{i2},\dots,F_{ij},\dots,F_{iq_i}\right)=P\left(X_i=k\middle|PA_{ij},F_{ij}\right)=f_{ijk}\\F_{ij}=\left(f_{ij1},f_{ij2},\dots,f_{ijr_i}\right)\end{gathered} \tag{4.3.5}$$

Note, every node $X_i$ has $r_i$ possible values (1, 2,…, $r_i$). Equation 4.3.5 is an extension of equation 4.1.5 for multi-node BN with Dirichlet density function.

The Dirichlet density function for each $F_{ij}$ is specified in equation 4.3.6 as follows:

$$\begin{aligned}\mathrm{Dir}\left(F_{ij}\right)&=\mathrm{Dir}\left(F_{ij}\middle|a_{ij1},a_{ij2},\dots,a_{ijr_i}\right)=\mathrm{Dir}\left(f_{ij1},f_{ij2},\dots,f_{ijr_i}\middle|a_{ij1},a_{ij2},\dots,a_{ijr_i}\right)\\&=\frac{\Gamma\left(N_{ij}\right)}{\prod_{k=1}^{r_i}a_{ijk}}\prod_{k=1}^{r_i}\left(f_{ijk}\right)^{a_{ijk}-1}\end{aligned} \tag{4.3.6}$$

Where,

$$N_{ij}=\sum_{k=1}^{r_i}a_{ijk}$$

$$a_{ijk}>0$$

$$\text{Each } f_{ijk} \text{ has respective parameter } a_{ijk}$$

Equation 4.3.6 is replication of equation 4.22. Variables $F_{ij}$ (s) attached to the same $X_i$ have no parent and are mutually independent, so, it is very easy to compute the joint Dirichlet density function Dir($F_{i1}, F_{i2},\ldots, F_{iq_i}$) with regard to node $X_i$ as follows:

$$\mathrm{Dir}(F_i) = \mathrm{Dir}\big(F_{i1}, F_{i2}, \dots, F_{ic_i}\big) = \mathrm{Dir}(F_{i1})\mathrm{Dir}(F_{i2}) \dots \mathrm{Dir}\big(F_{ic_i}\big) = \prod_{j=1}^{q_i} \mathrm{Dir}\big(F_{ij}\big) \qquad (4.3.7)$$

Besides the local parameter independence expressed in equation 4.3.7, we have global parameter independence if reviewing all variables *X<sub>i</sub>* (s) with note that all respective *F<sub>ij</sub>* (s) over entire augmented BN are mutually independent. Equation 4.3.8 expresses the global parameter independence of all *F<sub>ij</sub>* (s).

$$\mathrm{Dir}(F_1, F_2, \dots, F_i, \dots, F_n) = \mathrm{Dir}\begin{pmatrix} F_{11}, F_{12}, \dots, F_{1q_1}, F_{21}, F_{22}, \dots, F_{2q_2}, \dots, \\ F_{i1}, F_{i2}, \dots, F_{iq_i}, \dots, F_{n1}, F_{n2}, \dots, F_{nq_n} \end{pmatrix}$$
$$= \prod_{i=1}^{n} \mathrm{Dir}\big(F_{i1}, F_{i2}, \dots, F_{iq_i}\big) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \mathrm{Dir}\big(F_{ij}\big) \qquad (4.3.8)$$

Concepts “local parameter independence” and “global parameter independence” are defined in (Neapolitan, 2003, p. 333).

In trust BN, the conditional probability of variable $X_i$ with respect to its parent instance $PA_{ij}$, in other words, the $ij^{th}$ conditional distribution is expected value of $F_{ij}$ as below:

$$P\big(X_i = k \big| PA_{ij}\big) = E\big(f_{ijk}\big) = \frac{a_{ijk}}{N_{ij}} \qquad (4.3.9)$$

Equation 4.3.9 is extension of equation 4.1.9 and the proof of equation 4.3.9 is like the proof of equation 4.1.9.

Given multinomial sample $\mathcal{D} = (X^{(1)}, X^{(2)}, \dots, X^{(m)})$, equation 4.3.10 is used for calculating probability of evidences corresponding to variable $X_i$ over *m* trials as follows:

$$P\Big(X_i^{(1)}, X_i^{(2)}, \dots, X_i^{(M)} \Big| PA_i, F_i\Big) = \prod_{u=1}^{m} P\Big(X_i^{(u)} \Big| PA_i, F_i\Big) = \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \big(f_{ijk}\big)^{s_{ijk}} \qquad (4.3.10)$$

Where,

- Number $q_i$ is the number of parent instances of $X_i$.
- Counter $s_{ijk}$, respective to $F_{ij}$, is the number of all evidences among *m* trials such that variable $X_i = k$ given $PA_{ij}$.
- $PA_i = \{PA_{i1}, PA_{i2}, PA_{i3}, \dots, PA_{iq_i}\}$ is the vector of parent instances of $X_i$ and $F_i = \{F_{i1}, F_{i2}, \dots, F_{iq_i}\}$ is the respective vector of variables $F_{i1}$ (s) attached to $X_i$.

Equation 4.3.10 is extension of equation 4.1.10. From equation 4.3.10, we have equation 4.3.11 for calculating likelihood function $P(\mathcal{D}|F_1, F_2, \dots, F_n)$ of evidence sample $\mathcal{D}$ given *n* vectors $F_i$ (s).

$$P(\mathcal{D}|F_1, F_2, \dots, F_n) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \big(f_{ijk}\big)^{s_{ijk}} \qquad (4.3.11)$$

Please review equation 4.1.11 to know how to derive equation 4.3.11 because equation 4.3.11 is extension of equation 4.1.11. By extending equation 4.1.12, we get equation 4.3.12 to calculate marginal probability $P(\mathcal{D})$.

$$P(\mathcal{D}) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} E\left(\prod_{k=1}^{r_i} \big(f_{ijk}\big)^{s_{ijk}}\right) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma\big(N_{ij}\big)}{\Gamma\big(N_{ij} + M_{ij}\big)} \prod_{k=1}^{r_i} \frac{\Gamma\big(a_{ijk} + s_{ijk}\big)}{\Gamma\big(a_{ijk}\big)} \qquad (4.3.12)$$

Where,

$$M_{ij}=\sum_{k=1}^{r_i}s_{ijk}$$

Please make comparison among equations 4.1.12, 4.25, and 4.3.12 in order to comprehend that they share the same meaning. The proof of equation 4.3.12 is like the proof of equation 4.1.12.

Now, we need to compute posterior density function Dir($F_{ij}|\mathcal{D}$) and updated probability $P(X_i=k|PA_{ij}, \mathcal{D})$ for each variable $X_i$ in multi-node BN. By extending equation 4.1.13, we get equation 4.3.13 to calculate posterior density function Dir($F_{ij}|\mathcal{D}$).

$$\begin{aligned}\mathrm{Dir}\left(F_{ij}\middle|\mathcal{D}\right)&=\mathrm{Dir}\left(F_{ij}\middle|a_{ij1}+s_{ij1},a_{ij2}+s_{ij2},\dots,a_{ijr}++s_{ijr}\right)\\&=\frac{\Gamma\left(N_{ij}+M_{ij}\right)}{\prod_{k=1}^{r}\left(a_{ijk}+s_{ijk}\right)}\prod_{k=1}^{r_i}\left(f_{ijk}\right)^{a_{ijk}+s_{ijk}-1}\end{aligned}\quad(4.3.13)$$

Equation 4.3.13 is also replication of equation 4.26. The proof of equation 4.3.13 is like the proof of equation 4.1.13.

By extending equation 4.1.14, we get equation 4.3.14 to calculate updated probability $P(X_i=1|PA_{ij}, \mathcal{D})$.

$$P\left(X_i=k\middle|PA_{ij},\mathcal{D}\right)=E\left(f_{ijk}\middle|\mathcal{D}\right)=\frac{a_{ijk}+s_{ijk}}{N_{ij}+M_{ij}}\quad(4.3.14)$$

Equation 4.3.14 is also replication of equation 4.27. Please pay attention to equations 4.3.13 and 4.1.14 because they are main equations to determine posterior density function Dir($F_{ij}|\mathcal{D}$) and updated probability $P(X_i=k|PA_{ij}, \mathcal{D})$ for each variable $X_i$ in multi-node BN.

The concept of equivalent sample size, which is necessary to parameter learning, is also defined for multinomial sample learning. According to definition 4.3.1 (Neapolitan, 2003, p. 395), suppose there is a multinomial augmented BN and its parameters in full $\mathrm{Dir}\left(F_{ij}\middle|a_{ij1},a_{ij2},\dots,a_{ijr_i}\right)$, for all $i$ and $j$, if there exists the number $N$ such that satisfying equation 4.3.15 then, the multinomial augmented BN is called to have *equivalent sample size N*.

$$N_{ij}=\sum_{k=1}^{r_i}a_{ijk}=P\left(PA_{ij}\right)*N\quad(4.3.15)$$
$$(\forall i,j)$$

Where $P(PA_{ij})$ is probability of the $j^{th}$ parent instance of an $X_i$ and it is conventional that if $X_i$ has no parent then, $P(PA_{i1})=1$. If a multinomial augmented BN has equivalent sample size $N$ then, for each node $X_i$, we have:

$$\sum_{j=1}^{q_i}N_{ij}=\sum_{j=1}^{q_i}P\left(PA_{ij}\right)*N=N\sum_{j=1}^{q_i}P\left(PA_{ij}\right)=N$$

Where $q_i=\sum_{k=1}^{p_i}r_k$ is the number instances of parents of $X_i$. If $X_i$ has no parent then, $q_i=1$.

According to theorem 4.3.1 (Neapolitan, 2003, p. 396), suppose there is a multinomial augmented BN and its parameters in full $\mathrm{Dir}\left(F_{ij}\middle|a_{ij1},a_{ij2},\dots,a_{ijr_i}\right)$, for all $i$ and $j$, if there exists the number $N$ such that satisfying equation 4.3.16 then, the multinomial augmented BN has equivalent sample size $N$ and the embedded BN has uniform joint probability distribution.

$$a_{ijk}=\frac{N}{r_iq_i}\quad(4.3.16)$$

Where $q_i=\sum_{k=1}^{p_i}r_k$ is the number instances of parents of $X_i$. If $X_i$ has no parent then, $q_i=1$. It is easy to prove this theorem, we have:

$$\text{For all } i \text{ and } j, N_{ij} = \sum_{k=1}^{r_i} a_{ijk} = \frac{r_i N}{r_i q_i} = \frac{1}{q_i} N = P\left(PA_{ij}\right) * N$$

According to <u>theorem 4.3.2</u> (Neapolitan, 2003, p. 396), suppose there is a multinomial augmented BN and its parameters in full $\beta(F_{ij}; a_{ij}, b_{ij})$, for all $i$ and $j$, if there exists the number $N$ such that satisfying equation 4.3.17 then, the multinomial augmented BN has equivalent sample size $N$.

$$a_{ijk} = P\left(X_i = k \middle| PA_{ij}\right) * P\left(PA_{ij}\right) * N \tag{4.3.17}$$

Where $q_i = \sum_{k=1}^{p_i} r_k$ is the number instances of parents of $X_i$. If $X_i$ has no parent then, $q_i$=1. It is easy to prove this theorem, we have:

$$\text{For all } i \text{ and } j, N_{ij} = \sum_{k=1}^{r_i} a_{ijk} = \sum_{k=1}^{r_i} P\left(X_i = k \middle| PA_{ij}\right) * P\left(PA_{ij}\right) * N$$

$$= P\left(PA_{ij}\right) * N * \sum_{k=1}^{r_i} P\left(X_i = k \middle| PA_{ij}\right) = P\left(PA_{ij}\right) * N \blacksquare$$

According to <u>definition 4.3.2</u> (Neapolitan, 2003, p. 396), two multinomial augmented BNs: ($G_1$, $F^{(G1)}$, $\rho^{(G1)}$) and ($G_2$, $F^{(G2)}$, $\rho^{(G2)}$) are called *equivalent* (or *augmented equivalent*) if they satisfy following conditions:

1. $G_1$ and $G_2$ are Markov equivalent.
2. The probability distributions in their embedded BNs ($G_1$, $P_1$) and ($G_2$, $P_2$) are the same, $P_1 = P_2$.
3. Of course, $\rho^{(G1)}$ and $\rho^{(G2)}$ are Dirichlet distributions, $\rho^{(G1)}$ = Dir$^{(G2)}$ and $\rho^{(G2)}$ = Dir$^{(G2)}$.
4. They share the same equivalent size.

Note that we can make some mapping so that a node $X_i$ in ($G_1$, $F^{(G1)}$, Dir$^{(G1)}$) is also node $X_i$ in ($G_2$, $F^{(G2)}$, Dir$^{(G2)}$) and a parameter $F_i$ in ($G_1$, $F^{(G1)}$, Dir$^{(G1)}$) is also parameter $F_i$ in ($G_2$, $F^{(G2)}$, Dir$^{(G2)}$) if ($G_1$, $F^{(G1)}$, Dir$^{(G1)}$) and ($G_2$, $F^{(G2)}$, Dir$^{(G2)}$) are equivalent.

Given multinomial sample $\mathcal{D}$ and two multinomial augmented BNs ($G_1$, $F^{(G1)}$, $\rho^{(G1)}$) and ($G_2$, $F^{(G2)}$, $\rho^{(G2)}$), according to <u>lemma 4.3.1</u> (Neapolitan, 2003, p. 396), if such two augmented BNs are equivalent then, we have:

$$P_1(\mathcal{D}|G_1) = P_2(\mathcal{D}|G_1) \tag{4.3.18}$$

Where $P_1(\mathcal{D} \mid G_1)$ and $P_2(\mathcal{D} \mid G_2)$ are probabilities of sample $\mathcal{D}$ given parameters of $G_1$ and $G_2$, respectively. They are likelihood functions which are mentioned in equation 4.1.11.

$$P_1(\mathcal{D}|G_1) = P_1\left(\mathcal{D}\middle|F_1^{(G_1)}, F_2^{(G_1)}, \dots, F_n^{(G_1)}\right) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \left(f_{ijk}^{(G_1)}\right)^{s_{ijk}}$$

$$P_2(\mathcal{D}|G_2) = P_2\left(\mathcal{D}\middle|F_1^{(G_2)}, F_2^{(G_2)}, \dots, F_n^{(G_2)}\right) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \left(f_{ijk}^{(G_2)}\right)^{s_{ijk}}$$

Equation 4.3.18 specifies a so-called likelihood equivalence. In other words, if two augmented BNs are equivalent then, likelihood equivalence is obtained. Note, $f_{ijk}^{(G_l)}$ denotes parameter $f_{ijk}$ in BN ($G_l$, $P_l$).

According to <u>corollary 4.3.1</u> (Neapolitan, 2003, p. 397), given multinomial sample $\mathcal{D}$ and two multinomial augmented BNs ($G_1$, $F^{(G1)}$, $\rho^{(G1)}$) and ($G_2$, $F^{(G2)}$, $\rho^{(G2)}$), if such two augmented BNs are equivalent then, two updated probabilities corresponding two embedded BNs ($G_1$, $P_1$) and ($G_2$, $P_2$) are equal as follows:

$$P_1\left(X_i^{(G_1)}=1\middle|PA_{ij}^{(G_1)},\mathcal{D}\right)=P_2\left(X_i^{(G_2)}=1\middle|PA_{ij}^{(G_2)},\mathcal{D}\right) \quad (4.1.19)$$

These update probabilities are specified by equation 4.3.14.

$$P_1\left(X_i^{(G_1)}=k\middle|PA_{ij}^{(G_1)},\mathcal{D}\right)=E\left(F_{ij}^{(G_1)}\middle|\mathcal{D}\right)=\frac{a_{ijk}^{(G_1)}+s_{ijk}^{(G_1)}}{N_{ij}^{(G_1)}+M_{ij}^{(G_1)}}$$

$$P_2\left(X_i^{(G_2)}=k\middle|PA_{ij}^{(G_2)},\mathcal{D}\right)=E\left(F_{ij}^{(G_2)}\middle|\mathcal{D}\right)=\frac{a_{ijk}^{(G_2)}+s_{ijk}^{(G_2)}}{N_{ij}^{(G_2)}+M_{ij}^{(G_2)}}$$

Note, $X_i^{(G_l)}$ denotes node $X_i$ in $G_l$ and hence, other notations are similar.

# 5. Structure learning

As discussed in section 2, DAGs which have the same set of nodes are Markov equivalent if and only if they have same d-separations. From lemma 2.3.1 and theorem 2.3.2 (Neapolitan, 2003, pp. 86-87), Neapolitan (Neapolitan, 2003, p. 91) stated that Markov equivalence class can be represented with a graph that has the same links and the same uncoupled head-to-head meeting as the DAGs in the class. Markov equivalence divides all DAGs into disjoint Markov equivalence classes. According to (Neapolitan, 2003, p. 91), a DAG pattern is defined for a Markov equivalence class to be the graph that has the same links as the DAGs in the equivalence class and has oriented all and only the edges common to all DAGs in the equivalence class. According to theorem 2.4.2, if a joint probability distribution *P* is faithful to a DAG (admits a faithful DAG representation), there is a unique DAG pattern which is faithful to *P*. Therefore, although we cannot find out a unique DAG which is faithful to *P*, we can find out a unique DAG pattern which is faithful to *P*.

Let the pattern *gp* be a DAG pattern. Let *GP* be random variable whose values are patterns *gp*. The basic idea of structure learning is to find out the pattern *gp* that is faithful to *P*. There are two main learning structure approaches:

- *Score-based approach* (Neapolitan, 2003, pp. 441-476): For each pattern *gp* ∈ *GP*, the *gp* which gains the maximal scoring criterion *score*(*D*, *gp*) given training data set *D* is the best one. Because the essence of score-based approach is select the most likely structure based on the pre-defined *score*(*D*, *gp*), it is also called *model selection* approach (Neapolitan, 2003, p. 445).
- *Constraint-based approach* (Neapolitan, 2003, pp. 541-603): Given a set of conditional independences in a joint probability distribution, the best *gp* is the one for which Markov condition entails all and only these conditional independences. Such independences play the role of the “door latch” for learning algorithm. In other words, we try to find out the DAG that satisfies faithfulness condition (Neapolitan, 2003, p. 541).

## 5.1. Score-based approach

Given a set of random variables (nodes) $V = \{X_1, X_2,\ldots, X_n\}$, let (*G*, *P*) be possible BN where *P* is joint probability distribution and *G* = (*V*, *E*) is a DAG. Suppose (*G*, *P*) satisfies Markov condition and *P* is product of conditional probabilities (CPTs) of nodes give their parent according to theorem 2.1.2 (Neapolitan, 2003, p. 37). Let $(G, F^{(G)}, \beta^{(G)})$ be the augmented BN with equivalent sample size *N* where $F^{(G)}$ represents augmented variables attached to nodes in *V* and $\beta^{(G)}$ represents beta distributions for augmented variables (see section 4). Recall that, $(G, F^{(G)}, \beta^{(G)})$ is called the augmented BN of (*G*, *P*) is called the embedded BN of $(G, F^{(G)}, \beta^{(G)})$. Let *GP* be random variable whose values are DAG patterns *gp*. A DAG pattern (a value) *gp* represents a Markov equivalence class of DAGs including *G*. Thus, each *gp* corresponds with all equivalent augmented BNs ($G$, $F^{(G)}, \beta^{(G)}$) as well as all equivalent embedded BNs ($G$, $P$). When each CPT is calculated based on counter $s_{ij}$ and $t_{ij}$ with regard to augmented variables $F_{ij}$ according to equation 4.1.14 or equation 4.3.14, the joint probability distribution *P* is called relative frequency distribution. There are two so-called *score-based assumptions* for scored-based approach (Neapolitan, 2003, p. 442):

- Each relative frequency distribution *P* admits a faithful DAG representation. Exactly, *P* is faithful to its associated DAG pattern.
- After updating CPTs from multinomial sample according to equation 4.1.14 or equation 4.3.14 based on augmented BN, *P* is still faithful to its associated DAG pattern.

Given multinomial sampling with note that each $X_i$ is multinomial variable in general, scored-based approach has four following steps (Neapolitan, 2003, pp. 443-445):

1. Suppose we have a number of DAG patterns, $gp_1$, $gp_2$,…, $gp_K$ known as values of the random variable *GP*. For each $gp \in GP$, we construct a multinomial augmented BN ($G$, $F^{(G)}, \beta^{(G)}$) such that the DAG *G* is represented by *gp*. Each *gp* is also considered an event. Of course, we have such *K* multinomial augmented BNs. Note, ($G$, $F^{(G)}, \beta^{(G)}$) as well as ($G$, $P$) are assumed to satisfy faithfulness condition according to the first score-based assumption.
2. Let $r_i$ be the number of possible values of variable $X_i$. Note, in simpler case that $X_i$ is binary then, $r_i = 2$. Let $q_i$ be the number of distinct instantiations of parents of $X_i$. For example, if $X_i$ and its parents are binary and $X_i$ have 1 parent then $q_i = 2$. In general, if $X_i$ has a set of $p_i$ parent nodes and each parent node $X_k$ has $r_k$ possible values (1, 2,…, $r_k$), we have $q_i = \sum_{k=1}^{p_i} r_k$. Let $a_{ijk}$ be parameters of augmented variable $F_{ij}$ corresponding to the conditional probability of $X_i$ given instantiation *j* of its parent. According to theorem 4.3.1 (Neapolitan, 2003, p. 396), $a_{ijk}$ for each *gp* is initialized by equation 5.1.1 so that the respective augmented BN has the same equivalent sample size *N* and its embedded BN has uniform joint probability distribution.
$$a_{ijk} = \frac{N}{r_i q_i} \tag{5.1.1}$$
Note, *N* can be set arbitrarily. For convenience, equation 5.1.1 is replication of equation 4.3.16.
3. Given $\mathcal{D} = \{X^{(1)}, X^{(2)}, \ldots, X^{(M)}\}$ is the multinomial sample size *M*, where $X^{(u)}$ is a trial. Note that $X^{(u)}=(X_1^{(u)}, X_2^{(u)},\ldots, X_n^{(u)})$ is a *n*-dimension vector which is a outcome (instantiation) of *n* variables $X_i$, $X_2$,…, $X_n$. Each $X_i^{(u)}$ has the same space to $X_i$. Each DAG pattern *gp* is assigned a so-called scoring criterion *score*($\mathcal{D}$, *gp*). This score is the likelihood function of *gp* given training data set $\mathcal{D}$ (Neapolitan, 2003, p. 449).
$$score(\mathcal{D}, gp) = score(\mathcal{D}, G) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(N_{ij})}{\Gamma(N_{ij} + M_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(a_{ijk} + s_{ijk})}{\Gamma(a_{ijk})} \tag{5.1.2}$$
Where Γ(.) is gamma function specified by equation 4.13.
4. Suppose $\widehat{gp}$ maximizes *score*(*D*, *gp*) and then, a DAG $\hat{G}$ which is represented by $\widehat{gp}$ is selected as the resulted DAG. CPTs associated with nodes of $\hat{G}$ are determined by values $M_{ij}$ and $s_{ijk}$ based on Markov condition, according to equation 4.1.14 or equation 4.3.14. Please see section 4 for more details about parameter learning. The final joint probability distribution $\hat{P}$ is product of these CPTs. As a result, the learned BN $(\hat{G}, \hat{P})$ is determined

with expectation that such $(\hat{G}, \hat{P})$ satisfies faithfulness condition ($\hat{P}$ is faithful to $\hat{G}$), based on two score-based assumptions.

Recall that a DAG pattern is not a DAG because it is defined for a Markov equivalence class to be the graph that has the same links as the DAGs in the equivalence class and has oriented all and only the edges common to all DAGs in the equivalence class. Therefore, implicitly, equation 5.1.2 and other equations in this section apply any DAG that belongs to the Markov equivalence class represented by *gp* into their formulation.

Following is explanation of equation 5.1.2. The posterior probability of *gp* given data sample $\mathcal{D}$ is:

$$P(gp|\mathcal{D}) = \frac{P(\mathcal{D}|gp)P(gp)}{P(\mathcal{D})}$$

Where *P*(*gp*) is the prior probability and *P*($\mathcal{D}$|*gp*) is the likelihood function.

$$P(\mathcal{D}|gp) = \prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{\Gamma(N_{ij})}{\Gamma(N_{ij}+M_{ij})}\prod_{k=1}^{r_i}\frac{\Gamma(a_{ijk}+s_{ijk})}{\Gamma(a_{ijk})}$$

The best $\widehat{gp}$ is the one that maximizes such posterior probability *P*(*gp* | $\mathcal{D}$).

$$\widehat{gp} = \underset{gp}{\operatorname{argmax}}\, P(gp|\mathcal{D})$$

Because the marginal probability $P(\mathcal{D})$ is constant with regard to *gp*, the best $\widehat{gp}$ is now a maximizer of *P*($\mathcal{D}$ | *gp*)*P*(*gp*).

$$\widehat{gp} = \underset{gp}{\operatorname{argmax}}\, P(\mathcal{D}|gp)P(gp)$$

In practice, the prior probability *P*(*gp*) is uniform due to equation 5.1.1 and so *P*(*gp*) is ignored. Hence, the best $\widehat{gp}$ is a maximizer of the likelihood function *P*($\mathcal{D}$ | *gp*).

$$\widehat{gp} = \underset{gp}{\operatorname{argmax}}\, P(\mathcal{D}|gp)$$

The <u>scoring criterion</u> *score*($\mathcal{D}$, *gp*) is defined as such likelihood function *P*($\mathcal{D}$ | *gp*).

$$score(\mathcal{D}, gp) = score(\mathcal{D}, G) = P(\mathcal{D}|gp) = P(\mathcal{D}|G)$$
$$= \prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{\Gamma(N_{ij})}{\Gamma(N_{ij}+M_{ij})}\prod_{k=1}^{r_i}\frac{\Gamma(a_{ijk}+s_{ijk})}{\Gamma(a_{ijk})}$$

The likelihood function *P*($\mathcal{D}$ | *gp*) is the same for all Markov equivalent DAGs represented by *gp* given the same equivalent sample size because two equivalent augmented BNs have equal values of their likelihood functions according to lemma 4.1.1 (Neapolitan, 2003, p. 354) and lemma 4.3.1 (Neapolitan, 2003, p. 396). Note, *P*($\mathcal{D}$ | *gp*) which was formulated by equation 4.3.12 and so, equation 5.1.2 is replication of equation 4.3.12.

In case of binary sampling, due to $r_i$=2, equation 5.1.1 becomes:

$$a_{ij} = b_{ij} = \frac{N}{2q_i} \tag{5.1.3}$$

In case of binary sampling, due to $r_i$=2, equation 5.1.2 is degraded into equation 4.1.12.

$$score(\mathcal{D}, gp) = score(\mathcal{D}, G) = P(\mathcal{D}|gp) = P(\mathcal{D}|G)$$
$$= \prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{\Gamma(N_{ij})}{\Gamma(N_{ij}+M_{ij})}\frac{\Gamma(a_{ij}+s_{ij})\Gamma(b_{ij}+t_{ij})}{\Gamma(a_{ij})\Gamma(b_{ij})} \tag{5.1.4}$$

For convenience, equations 5.1.3 and 5.1.4 are replication of equations 4.1.16 and 4.1.12, respectively.

**Example 5.1.1.** Suppose there are two binary variables $X_1$, $X_2$, we don't know exactly their relationship but the binomial sample $\mathcal{D}$ is observed as seen in table 5.1.1 (Neapolitan, 2003, p. 446).

| $X_1$ | $X_2$ |
|---|---|
| 1 | 1 |
| 1 | 0 |
| 1 | 1 |
| 0 | 0 |
| 1 | 1 |
| 0 | 1 |
| 1 | 1 |
| 0 | 0 |

**Table 5.1.1.** Binomial sample for illustrating score-based approach

Let $gp_1$ be the DAG pattern in which $X_1$ is parent of $X_2$; otherwise let $gp_2$ be the DAG pattern in which $X_1$ and $X_2$ are mutually independent. Given equivalent sample size is $N = 4$, figure 5.1.1 shows such $gp_1$ and $gp_2$. Note, the DAG $X_1 \rightarrow X_2$ in figure 5.1.1 (a) is a member of Markov equivalence class represented by $gp_1$ and so $gp_1$ also represents another DAG $X_1 \leftarrow X_2$. Hence, $gp_1$ represents no conditional independence whereas $gp_2$ represents the conditional independence $I_P(\{X_1\}, \{X_2\})$ (Neapolitan, 2003, p. 443).

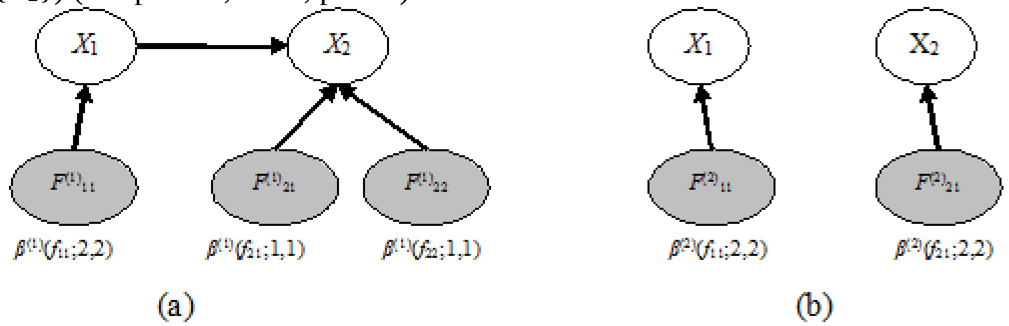

$X_1$
$X_2$
$F^{(1)}_{11}$
$F^{(1)}_{21}$
$F^{(1)}_{22}$
$\beta^{(1)}(f_{11};2,2)$
$\beta^{(1)}(f_{21};1,1)$
$\beta^{(1)}(f_{22};1,1)$
(a)
$X_1$
$X_2$
$F^{(2)}_{11}$
$F^{(2)}_{21}$
$\beta^{(2)}(f_{11};2,2)$
$\beta^{(2)}(f_{21};2,2)$
(b)

**Figure 5.1.1.** Augmented Bayesian networks of $gp_1$ in (a) and $gp_2$ in (b)

For $gp_1$, given equivalent sample size is $N = 4$, according to equation 5.1.3, we have $N_{11}^{(1)} = 4$, $N_{21}^{(1)} = 2$, $N_{22}^{(1)} = 2$, $a_{11}^{(1)} = b_{11}^{(1)} = 2$, $a_{21}^{(1)} = b_{21}^{(1)} = 1$, and $a_{22}^{(1)} = b_{22}^{(1)} = 1$. From table 5.1.1, there are 5 cases of $X_1 = 1$ and 3 cases of $X_1 = 0$; hence we have $M_{11}^{(1)} = 8$, $s_{11}^{(1)} = 5$ and $t_{11}^{(1)} = 3$. From table 5.1.1, given $X_1 = 1$, there are 4 cases of $X_2 = 1$ and 1 case of $X_2 = 0$; hence we have $M_{21}^{(1)} = 5$, $s_{21}^{(1)} = 4$ and $t_{21}^{(1)} = 1$. From table 5.1.1, given $X_1 = 0$, there are 1 case of $X_2 = 1$ and 2 cases of $X_2 = 0$; hence we have $M_{22}^{(1)} = 3$, $s_{22}^{(1)} = 1$ and $t_{22}^{(1)} = 2$. According to equation 5.1.4, the score of $gp_1$ is (Neapolitan, 2003, p. 446):

$$\begin{aligned}
score(\mathcal{D}, gp_1) = & \left(\frac{\Gamma\left(N_{11}^{(1)}\right)}{\Gamma\left(N_{11}^{(1)} + M_{11}^{(1)}\right)} \frac{\Gamma\left(a_{11}^{(1)} + s_{11}^{(1)}\right)\Gamma\left(b_{11}^{(1)} + t_{11}^{(1)}\right)}{\Gamma\left(a_{11}^{(1)}\right)\Gamma\left(b_{11}^{(1)}\right)}\right) \\
& * \left(\frac{\Gamma\left(N_{21}^{(1)}\right)}{\Gamma\left(N_{21}^{(1)} + M_{21}^{(1)}\right)} \frac{\Gamma\left(a_{21}^{(1)} + s_{21}^{(1)}\right)\Gamma\left(b_{21}^{(1)} + t_{21}^{(1)}\right)}{\Gamma\left(a_{21}^{(1)}\right)\Gamma\left(b_{21}^{(1)}\right)}\right) \\
& * \left(\frac{\Gamma\left(N_{22}^{(1)}\right)}{\Gamma\left(N_{22}^{(1)} + M_{22}^{(1)}\right)} \frac{\Gamma\left(a_{22}^{(1)} + s_{22}^{(1)}\right)\Gamma\left(b_{22}^{(1)} + t_{22}^{(1)}\right)}{\Gamma\left(a_{22}^{(1)}\right)\Gamma\left(b_{22}^{(1)}\right)}\right)
\end{aligned}$$

$$= \left(\frac{\Gamma(4)}{\Gamma(4+8)}\frac{\Gamma(2+5)\Gamma(2+3)}{\Gamma(2)\Gamma(2)}\right) * \left(\frac{\Gamma(2)}{\Gamma(2+5)}\frac{\Gamma(1+4)\Gamma(1+1)}{\Gamma(1)\Gamma(1)}\right)$$
$$* \left(\frac{\Gamma(2)}{\Gamma(2+3)}\frac{\Gamma(1+1)\Gamma(1+2)}{\Gamma(1)\Gamma(1)}\right)$$

Note, Γ(2) = Γ(1) = 1 and the gamma function Γ(*x*) degrades into factorial function (*x*–1)! when *x* is an integer. Hence, we have (Neapolitan, 2003, p. 446):

$$score(\mathcal{D}, gp_1) = \frac{\Gamma(3)\Gamma(4)\Gamma(5)}{\Gamma(12)} = \frac{2!\,3!\,4!}{11!} \approx 7.2150 * 10^{-6}$$

For $gp_2$, given equivalent sample size is $N = 4$, according to equation 5.1.3, we have $N_{11}^{(2)} = 4$, $N_{21}^{(2)} = 4$, $N_{22}^{(2)} = 4$, $a_{11}^{(2)} = b_{11}^{(2)} = 2$, and $a_{21}^{(2)} = b_{21}^{(2)} = 2$. From table 5.1.1, there are 5 cases of $X_1 = 1$ and 3 cases of $X_1 = 0$; hence we have $M_{11}^{(2)} = 8$, $s_{11}^{(2)} = 5$ and $t_{11}^{(2)} = 3$. From table 5.1.1, there are 5 cases of $X_2 = 1$ and 3 cases of $X_2 = 0$; hence we have $M_{21}^{(2)} = 8$, $s_{21}^{(2)} = 5$ and $t_{21}^{(2)} = 3$. According to equation 5.1.4, the score of $gp_2$ is (Neapolitan, 2003, p. 446):

$$score(\mathcal{D}, gp_2) = \left(\frac{\Gamma\left(N_{11}^{(2)}\right)}{\Gamma\left(N_{11}^{(2)} + M_{11}^{(2)}\right)}\frac{\Gamma\left(a_{11}^{(2)} + s_{11}^{(2)}\right)\Gamma\left(b_{11}^{(2)} + t_{11}^{(2)}\right)}{\Gamma\left(a_{11}^{(2)}\right)\Gamma\left(b_{11}^{(2)}\right)}\right)$$
$$* \left(\frac{\Gamma\left(N_{21}^{(2)}\right)}{\Gamma\left(N_{21}^{(2)} + M_{21}^{(2)}\right)}\frac{\Gamma\left(a_{21}^{(2)} + s_{21}^{(2)}\right)\Gamma\left(b_{21}^{(2)} + t_{21}^{(2)}\right)}{\Gamma\left(a_{21}^{(2)}\right)\Gamma\left(b_{21}^{(2)}\right)}\right)$$
$$= \left(\frac{\Gamma(4)}{\Gamma(4+8)}\frac{\Gamma(2+5)\Gamma(2+3)}{\Gamma(2)\Gamma(2)}\right) * \left(\frac{\Gamma(4)}{\Gamma(4+8)}\frac{\Gamma(2+5)\Gamma(2+3)}{\Gamma(2)\Gamma(2)}\right)$$

Note, Γ(2) = Γ(1) = 1 and the gamma function Γ(*x*) degrades into factorial function (*x*–1)! when *x* is an integer. Hence, we have (Neapolitan, 2003, p. 446):

$$score(\mathcal{D}, gp_2) = \left(\frac{\Gamma(4)\Gamma(5)\Gamma(7)}{\Gamma(12)}\right)^2 = \left(\frac{3!\,4!\,6!}{11!}\right)^2 \approx 6.7465 * 10^{-6}$$

Because the *score*($\mathcal{D}$, $gp_1$) is larger than the *score*($\mathcal{D}$, $gp_2$), the DAG pattern $gp_1$ is selected. Therefore the DAG $G^{(1)}$ shown in figure 5.1.1 (a), which is in Markov equivalence class represented by $gp_1$ is the resulted DAG. So, the resulted (embedded) BN appropriate to the binomial sample shown in table 5.1.1 is ($G^{(1)}$, $P^{(1)}$). According to equation 4.1.14, CPTs associated with $G^{(1)}$ are computed as follows (Neapolitan, 2003, p. 447):

$$P^{(1)}(X_1 = 1) = \frac{a_{11}^{(1)} + s_{11}^{(1)}}{N_{11}^{(1)} + M_{11}^{(1)}} = \frac{2+5}{4+8} \approx 0.58$$

$$P^{(1)}(X_2 = 1|X_1 = 1) = \frac{a_{21}^{(1)} + s_{21}^{(1)}}{N_{21}^{(1)} + M_{21}^{(1)}} = \frac{1+4}{2+5} \approx 0.71$$

$$P^{(1)}(X_2 = 1|X_1 = 0) = \frac{a_{22}^{(1)} + s_{22}^{(1)}}{N_{22}^{(1)} + M_{22}^{(1)}} = \frac{1+1}{2+3} \approx 0.40$$

The joint probability distribution $P^{(1)}$ is product of these CPTs. Figure 5.1.2 shows the resulted ($G^{(1)}$, $P^{(1)}$) from the scored-based approach■

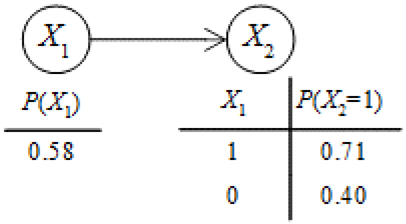


**Figure 5.1.2.** A resulted BN from scored-based approach

In example above we recognize that it is difficult to determine all DAG patterns. So the score-based approach becomes ineffective in case of many variables. The number of DAG pattern which is surveyed to compute scoring criterion gets huge. It is impossible to do brute-force searching over DAG space consisting of all possible DAGs. There are some heuristic algorithms to reduce whole DAG space to smaller space called candidate set of DAGs obeying some restriction, for example, the prior ordering of variables. Such heuristic algorithms are classified into *approximate learning* or approximate selection (Neapolitan, 2003, pp. 511-538). The global score *score*($\mathcal{D}$, *gp*) can defined as a product of local scores (Neapolitan, 2003, p. 512):

$$score(\mathcal{D},gp)=score(\mathcal{D},G)=\prod_{i=1}^{n}score(\mathcal{D},X_i,PA_i) \tag{5.1.5}$$

Where *score*(*D*, $X_i$, $PA_i$) is the local score of $X_i$ given its parents $PA_i$ and *n* is the number of nodes. The local score is defined by equation 5.1.6 (Neapolitan, 2003, p. 512).

$$score(\mathcal{D},X_i,PA_i)=\prod_{j=1}^{q^{(PA_i)}}\left(\frac{\Gamma\left(\sum_k a_{ijk}^{(PA_i)}\right)}{\Gamma\left(\sum_k a_{ijk}^{(PA_i)}+\sum_k s_{ijk}^{(PA_i)}\right)}\prod_{k=1}^{r_i}\frac{\Gamma\left(a_{ijk}^{(PA_i)}+s_{ijk}^{(PA_i)}\right)}{\Gamma\left(a_{ijk}^{(PA_i)}\right)}\right) \tag{5.1.6}$$

Where $q^{(PA_i)}$ is the number of distinct instantiations of variables in $PA_i$ whereas $s_{ijk}^{(PA_i)}$ is the number of cases in which $X_i = k$ and variables in $PA_i$ are in their *j*th instantiation, etc. Note that $a_{ijk}^{(PA_i)}$ depends only on *i*, *j*, *k*, and $PA_i$ and so it does not depend on the whole DAG.

A well-known heuristic algorithm belonging to approximate learning is K2 algorithm developed by Cooper and Herskovits in 1992. The K2 algorithm tries to find out the DAG whose each variable $X_i$ maximizes local score *score*($\mathcal{D}$, $X_i$, $PA_i$) instead of discovering all DAGs. It means that K2 algorithm finds out optimal parents $PA_i$ of each $X_i$. Note that it expects that the global score will be approached by maximizing each partial local score. K2 algorithm has following steps (Neapolitan, 2003, p. 513):

1. Suppose there is an ordering ($X_1$, $X_2$,…, $X_n$). There is no backward edge, for example, the edge $X_i \leftarrow X_j$ (if exist) where $i < j$ is invalid. Let *Pred*($X_i$) be the set of previous nodes of $X_i$ in the ordering. Let $PA_i$ is parents of $X_i$. K2's mission is to find out $PA_i$ for every $X_i$. Firstly, each $PA_i$ (s) is set to be empty and each local score *score*($\mathcal{D}$, $X_i$, $PA_i$) is initialized with such empty $PA_i$.
2. Each $X_i$ is visited according to the ordering. When $X_i$ is visited, which node in *Pred*($X_i$) that maximizes the local *score*($\mathcal{D}$, $X_i$, $PA_i$) is added to $PA_i$.
3. Algorithm terminates when no node is added to $PA_i$.

Following is C-like pseudo-code of K2 algorithm (Neapolitan, 2003, pp. 513-514).

Inputs: data sample $\mathcal{D}$, an ordering $p$ = ($X_1$, $X_2$,…, $X_n$), an upper bound *u* on the number of parents a node may have.

Outputs: *n* sets of parent nodes $PA_i$ where $1 \leq i \leq n$.

```
void $K2$ (data-sample $\mathcal{D}$, ordering $p$, int $u$, for $1 \leq i \leq n$ parent_set& $PA_i$)
{
    for ($i$ = 1; $i$ <= $n$; $i$ ++) { // $n$ is the number of nodes.
        $PA_i = Ø$;
        $P_{old} = score(\mathcal{D}, X_i, PA_i)$;
        $find\_more$ = true;
        while ($find\_more$ && $|PA_i| < u$) {
            $Z$ = nodes in $Pred(X_i) \setminus PA_i$ that maximizes $score(\mathcal{D}, X_i, PA_i \cup \{Z\})$;
            $P_{new} = score(\mathcal{D}, X_i, PA_i \cup \{Z\})$;
            if ($P_{new} > P_{old}$) {
                $P_{old} = P_{new}$;
                $PA_i = PA_i \cup \{Z\}$;
            }
            else
                $find\_more$ = false;
        }
    }
}
```

Note, the sign "\" denotes the subtraction (excluding) in set theory (Wikipedia, Set (mathematics), 2014). A weak point of K2 algorithm is to pre-define an ordering. In practice, such ordering is based on knowledge base, for example, smoking precedes bronchitis and lung cancer in medical diagnosis (Neapolitan, 2003, p. 515). If there is no prior knowledge domain, it is necessary to establish an algorithm which does not require prior ordering. Such heuristic algorithm is called *DAG-search algorithm*.

Let DAGOPS be set of operations defined as follows (Neapolitan, 2003, p. 516):

1. If two nodes are not adjacent, adding an edge between them in either direction.
2. If two nodes are adjacent, removing the edge between them.
3. If two nodes are adjacent, reversing the edge between them.

It requires that any DAGOPS operation does not produce cycle in DAG. Given a DAG *G*, the set of all DAGs obtaining from *G* by applying one of DAGOPS operations (one DAGOPS operation type) is called neighbor-set of G denoted *ns*(*G*). Of course, if *G*' belongs to *ns*(*G*) then, *G*' is called a neighbor of *G*. The set DAGOPS is complete because given two *G* and *G*' there is always a sequence of operations in DAGOPS that transform *G* into *G*'.

The ideology of DAG-search algorithm is simple. It starts with a DAG *G* without edges. At each step of the search, among neighbors of the current DAG *G*, it selects the nearest neighbor *G*' that maximizes the global score $score(\mathcal{D}, gp)$ where neighbors replace *gp* in formulation of $score(\mathcal{D}, gp)$. The current *G* is modified by applying DAGOPS operations into *G* so that *G* = *G*'. In other words, the nearest neighbor becomes the current DAG. Algorithm stops when there is no DAGOPS operation that increases $score(\mathcal{D}, gp)$. Note that in each step, if an edge to $X_i$ is added or deleted, we need only re-evaluate $score(\mathcal{D}, X_i, PA_i)$. If an edge between $X_i$ and $X_j$ is reversed, we need only re-evaluate $score(\mathcal{D}, X_i, PA_i)$ and $score(\mathcal{D}, X_j, PA_j)$. Therefore, computational cost for $score(\mathcal{D}, gp)$ is decreased significantly because $score(\mathcal{D}, gp)$ is product of many $score(\mathcal{D}, X_i, PA_i)$ according to equation 5.1.6. Moreover, in each step, each neighbor is created by modifying one edge for a DAGOPS operation type (adding, removing, or reversing). Following is C-like pseudo-code of DAG-search algorithm (Neapolitan, 2003, p. 517).

Inputs: data sample $\mathcal{D}$.

Outputs: A set of edges $E$ in a DAG that approximates maximizing $score(\mathcal{D}, gp)$.

```
void DAG-search(data-sample $\mathcal{D}$, set_of_edges& $E$)
{
    $E$ = Ø; $G = (V, E)$;
    while (some DAGOPS operation increases $score(\mathcal{D}, gp)$) {
        Let $G'$ be the nearest of $G$;
        Modify $E$ by DAGOPS operation so that $G = G'$;
        Update $score(\mathcal{D}, X_i, PA_i)$;
    }
}
```

Note, both K2 algorithm and DAG-search algorithm search for DAGs instead DAG patterns. There are other algorithms that search for DAG patterns mentioned in (Neapolitan, 2003, pp. 518-529).

## 5.2. Constraint-based approach

Given a BN ($G$, $P$) and $G = (V, E)$, let $IND_P$ be a set of conditional independences based on $P$. We assume that contains all and only entailed conditional independences. According to theorem 2.2.1 in (Neapolitan, 2003, p. 76), based on the Markov condition, a DAG $G$ entails all and only (entailed) conditional independencies that are identified by d-separations in $G$. Because it is impossible to distinguish DAGs in the same Markov equivalence class from $IND_P$, we can only learn the DAG pattern whose d-separations are the same as $IND_P$ (Neapolitan, 2003, p. 541). Therefore, constraint-based approach tries to find a DAG pattern whose d-separations are the same as $IND_P$ with suppose that $P$ admits a faithful DAG representation.

When learning DAG structure, we know neither $IND_P$ nor d-separations in $G$ except that faithfulness condition is assumed to be satisfied. Hence, we assume that there is a set of d-separations $IND$, for example $IND = \{I(\{X\}, \{Y\}), I(\{X\}, \{Y\}|\{Z\})\}$. The subscript $G$ is removed from d-separation notation because we do not determine structure of DAG $G$ yet. Hence, $IND$ is set of statements. Of course, it is assumed that $IND$ admits a faithful DAG representation because we expect that $IND$ is the same to $IND_P$ when it is supposed that $P$ admits a faithful DAG representation. In other words, $IND$ is assumed to contains all and only d-separations. In general, $IND$ is considered as set of constraints and so constraint-based approach tries to find DAG pattern that satisfies $IND$.

Lemma 2.3.2 (Neapolitan, 2003, p. 91), lemma 2.3.3 (Neapolitan, 2003, p. 91), and theorem 2.4.3 (Neapolitan, 2003, p. 99) are mainly used for constraint-based approach.

Recall that

[

According to lemma 2.3.2 (Neapolitan, 2003, p. 91), let $gp$ be DAG pattern and $X$ and $Y$ be nodes in $gp$ then, $X$ and $Y$ are adjacent in $gp$ if and only if they are not d-separated by some set in $gp$. According to lemma 2.3.3 (Neapolitan, 2003, p. 91), suppose we have a DAG pattern $gp$ and an uncoupled meeting $X–Z–Y$ then, the three followings are equivalent:

1. $X–Z–Y$ is a head-to-head meeting.
2. There exists a set not containing $Z$ that d-separates $X$ and $Y$.
3. All sets containing $Z$ do not d-separate $X$ and $Y$.

According to theorem 2.4.3 (Neapolitan, 2003, p. 99), suppose a joint probability distribution $P$ admits some faithful DAG representation then, $gp$ is the DAG pattern faithful to $P$ if and only if the two following conditions are satisfied:

1. $X$ and $Y$ are adjacent in *gp* if and only if there is no subset $S \subseteq V$ such that $I_P(\{X\}, \{Y\} \mid S)$ holds. That is, $X$ and $Y$ are adjacent if and only if there is a direct dependence between $X$ and $Y$.
2. Any chain $X-Z-Y$ is a head-to-head meeting in *gp* if and only if $Z \in S$ implies $NI_P(\{X\}, \{Y\} \mid S)$).

]

**Example 5.2.1**. Suppose we have $V = \{X, Y, Z\}$ and $IND = \{I(\{X, Y\} \mid \{Z\})\}$ (Neapolitan, 2003, p. 543). Because *IND* is assumed to contains all and only d-separations whereas $X$ and $Z$ are not *d*-separated from any set, there must be a link between $X$ and $Z$ according to lemma 2.3.2 (Neapolitan, 2003, p. 91). In similar, there is must be a link between $Y$ and $Z$ as shown in figure 5.2.1 (a).

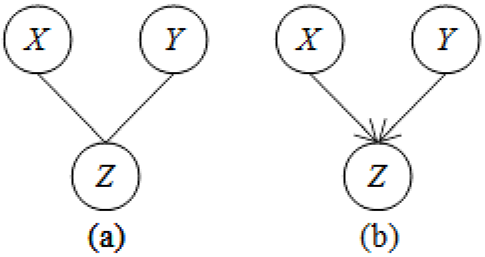


**Figure 5.2.1.** An example of constraint-based approach

Because *IND* is assumed to contains all and only d-separations whereas $X–Z–Y$ is uncoupled meeting and there is the d-separation $I(\{X, Y\} \mid \{Z\})$, the meeting $X–Z–Y$ must be head-to-head meeting $X \rightarrow Z \leftarrow Y$ according to lemma 2.3.3 (Neapolitan, 2003, p. 91) as shown in figure 5.2.1 (b)■

If the number of variables is large, we need effective algorithms. A simple algorithm called *find-DAG-pattern algorithm* includes two steps:

1. Firstly, the structure of DAG is drafted as "skeleton". If there is no conditional independence relating to $X_i$ and $X_j$ then the link between them is created. So skeleton is the undirected graph which contains variables (nodes) and links. This step follows lemma 2.3.2 (Neapolitan, 2003, p. 91).
2. The second step is to determine direction of links by applying four following rules in sequence rule 1, rule 2, rule 3, rule 4 (Neapolitan, 2003, p. 547):
   - *Rule* 1: If uncoupled meeting $X–Z–Y$ exists and $Z$ is not in any set that d-separate $X$ from $Y$ then, this meeting is converted as head-to-head meeting: $X \rightarrow Z \leftarrow Y$. This step follows lemma 2.3.3 (Neapolitan, 2003, p. 91).
   - *Rule* 2: If the uncoupled meeting $X \rightarrow Z–Y$ exists (having an edge $X \rightarrow Z$) then, this meeting is converted as head-to-tail meeting: $X \rightarrow Z \rightarrow Y$.
   - *Rule* 3: If an possible edge $X \rightarrow Y$ can cause a directed cycle at a position in network then it is reversed: $X \leftarrow Y$. This rule is applied to remove directed cycles so that the expected BN is a DAG.
   - *Rule* 4: If all rules 1, 2, and 3 are consumed, all remaining links have arbitrary direction. This rule is not necessary to apply into find-DAG-pattern algorithm because a DAG pattern can have links but rule 4 is useful to convert the resulted DAG pattern into a real DAG.

**Example 5.2.2**: Suppose we have $V = \{X, Y, Z, T\}$ and $IND = \{I(X,Y), I(X,T), I(Y,T)\}$. Because there is no conditional independence between $X$ and $Y$, between $Z$ and $T$, the "skeleton" is drafted as seen in figure 5.2.2 (a).

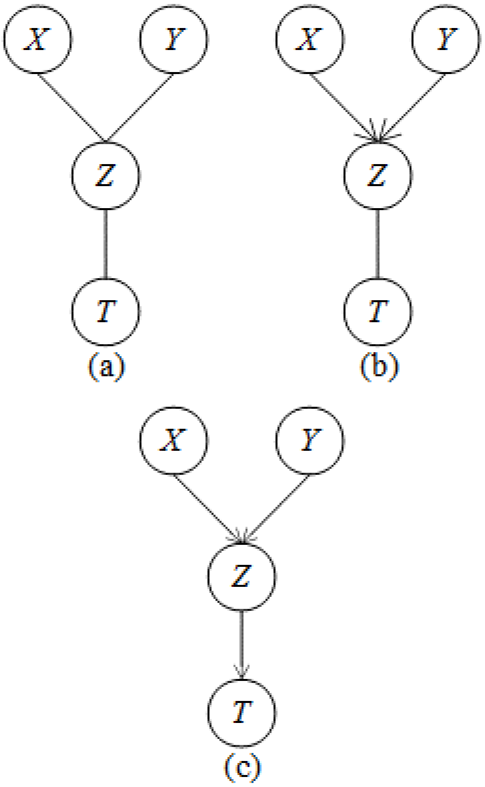


**Figure 5.2.2.** An example of find-DAG-pattern algorithm

By applying rule 1, because the uncoupled meeting *X*–*Z*–*Y* exists and *Z* is not in any set that d-separate *X* from *Y*, this meeting become head-to-head meeting *X*→*Z*←*Y* as seen in figure 5.2.2 (b). By applying rule 2, because the uncoupled meeting *X*→*Z*–*T* exists, we have and head-to-tail meeting *X*→*Z*→*T* as seen in figure 5.2.2 (c)■

Following is C-like pseudo-code of find-DAG-pattern algorithm (Neapolitan, 2003, p. 517).

Inputs: a set *V* of nodes and a set *IND* of d-separations.

Outputs: DAG pattern *gp* containing d-separations in I*ND*.

```
void find-DAG-pattern(set-of-nodes V, set-of-d-separations IND, DAG-pattern& gp)
{
    // Step 1
    for (each pair of nodes $X$, $Y \in$ V) {
        search for a subset $S_{XY} \subseteq$ V such that $I(\{X\}, \{Y\} \mid S_{XY})$;
        if (no such set can be found) create the link $X-Y$ in gp;
    }

    // Step 2
    for (each uncoupled meeting $X-Z-Y$) {
        if (Z $\notin S_{XY}$)
            orient $X-Z-Y$ as $X \rightarrow Z \leftarrow Y$; // Apply rule 1
    }
    while (more edges can be oriented) {
        for (each uncoupled meeting $X \rightarrow Z-Y$) {
            orient $Z-Y$ as $Z \rightarrow Y$; // Apply rule 2
        }
        for (each link $X–Y$ such that there is a path from $X$ to $Y$) {
            orient $X−Y$ as $X\rightarrow Y$; // Apply rule 3 to avoid directed cycle
        }
        for (each uncoupled meeting $X−Z–Y$
            such that $X\rightarrow W$, $Y\rightarrow W$, and $Z−W$) {
            orient $Z−W$ as $Z\rightarrow W$; // Apply rule 3 to avoid directed cycle
        }
    }
}
```

# 6. Conclusions

Three significant domains of Bayesian network (BN) are inference mechanism, parameter learning and structure learning. The first domain tells the usability of BN and the others indicates how to build up BN. The ideology of BN is to apply a mathematical inference tool (namely Bayesian rule) into a graph with expectation of extending and enhancing ability of such tool so as to solve realistic problems, especially diagnosis domain.

However, in process of developing BN, there are many problems involving in real number (continuous case) and nodes dependence. This report focuses on discrete case when probability of each node is discrete CPT, not continuous PDF. The first-order Markov condition has important role in BN study when there is an assumption “nodes are dependent on only their direct parents”. If the first-order Markov condition isn’t satisfied, many inference and learning algorithms go wrong. I think that BN will get more potential and enjoyable if first-order (Markov) condition is replaced by $n^{\text{th}}$-order condition.

Moreover, parameter and structure learning become difficult when training data is missing (not complete). Missing data problem is introduced in section 3 but its detail goes beyond this report. I hope that I have a chance to research it.

Finally, BN discussed here is “static” BN because temporal relationships among nodes are not concerned. The “static” BN is represented at only one time point. Otherwise dynamic Bayesian network (DBN) aims to model the temporal relationships among nodes. The process of inference is concerned in time series; in some realistic case this is necessary. However, the cost of inference and learning in DBN is much higher than BN because the size of DBN gets huge for long-time process. Because of the limitation of this report, the algorithm that keeps the size of DBN intact (not changed) is not introduced here. In general, the essence of such algorithm is to take advantage of both Markov condition and knowledge (inference) accumulation. Due to complexity of DBN, we should consider choosing which one (BN or DBN) to apply into concrete domain. It depends on what your domain is and what your purpose is.

# References

Borman, S. (2004). *The Expectation Maximization Algorithm - A short tutorial.* University of Notre Dame, Department of Electrical Engineering. South Bend, Indiana: Sean Borman's Home Page.

Fenton, N. E., Noguchi, T., & Neil, M. (2019, January 10). An Extension to the Noisy-OR Function to Resolve the ‘Explaining Away’ Deficiency for Practical Bayesian Network Problems. (X. Lin, Ed.) *IEEE Transactions on Knowledge and Data Engineering (TKDE), 31*(12), 2441 - 2445. doi:10.1109/TKDE.2019.2891680

Heckerman, D. (1995). *A Tutorial on Learning With Bayesian Networks.* Microsoft Corporation, Microsoft Research. Redmond: Microsoft Research. Retrieved from ftp://ftp.research.microsoft.com/pub/dtg/david/tutorial.ps

Montgomery, D. C., & Runger, G. C. (2003). *Applied Statistics and Probability for Engineers* (3rd Edition ed.). New York, NY, USA: John Wiley & Sons, Inc.

Murphy, K. P. (1998). *A Brief Introduction to Graphical Models and Bayesian Networks.* Retrieved 2008, from Kevin P. Murphy's home page: http://www.cs.ubc.ca/~murphyk/Bayes/bnintro.html

Neapolitan, R. E. (2003). *Learning Bayesian Networks.* Upper Saddle River, New Jersey, USA: Prentice Hall.

Nguyen, L. (2014, April). *A User Modeling System for Adaptive Learning.* University of Science, Ho Chi Minh city, Vietnam. Abuja, Nigeria: Standard Research Journals. Retrieved from http://standresjournals.org/journals/SSRE/Abstract/2014/april/Loc.html

Pearl, J. (1986, September). Fusion, propagation, and structuring in belief networks. *Artificial Intelligence, 29*(3), 241-288. doi:10.1016/0004-3702(86)90072-X

Rosen, K. H. (2012). *Discrete Mathematics and Its Applications* (7nd Edition ed.). (M. Lange, Ed.) McGraw-Hill Companies.

Shachter, R. D., D'Ambrosio, B., & Del Favero, B. A. (1990). Symbolic Probabilistic Inference in Belief Networks. In H. Shrobe (Ed.), *The Eighth National Conference on Artificial Intelligence (AAAI-90). 90*, pp. 126-131. Boston: The Association for the Advancement of Artificial Intelligence. Retrieved from http://www.aaai.org/Papers/AAAI/1990/AAAI90-019.pdf

Walpole, R. E., Myers, R. H., Myers, S. L., & Ye, K. (2012). *Probability & Statistics for Engineers & Scientists* (9th ed.). (D. Lynch, Ed.) Boston, Massachusetts, USA: Pearson Education, Inc.

Wikipedia. (2006). *Bayesian inference*. (Wikimedia Foundation) Retrieved 2007, from Wikipedia website: http://en.wikipedia.org/wiki/Bayesian_inference

Wikipedia. (2014, October 29). *Sample (statistics)*. (Wikimedia Foundation) Retrieved October 31, 2014, from Wikipedia website: http://en.wikipedia.org/wiki/Sample_(statistics)

Wikipedia. (2014, October 10). *Set (mathematics)*. (A. Rubin, Editor, & Wikimedia Foundation) Retrieved October 11, 2014, from Wikipedia website: http://en.wikipedia.org/wiki/Set_(mathematics)

Wikipedia. (2017, March 2). *Maximum a posteriori estimation*. (Wikimedia Foundation) Retrieved April 15, 2017, from Wikipedia website: https://en.wikipedia.org/wiki/Maximum_a_posteriori_estimation

Wikipedia. (2018, January 15). *Conjugate prior*. (Wikimedia Foundation) Retrieved February 15, 2018, from Wikipedia website: https://en.wikipedia.org/wiki/Conjugate_prior

Zhu, W. (2018). *Bayesian Inference for the Normal Distribution.* Stony Brook University, Department of Applied Mathematics and Statistics. New York: Stony Brook University. Retrieved October 21, 2019, from http://www.ams.sunysb.edu/~zhu/ams570/Bayesian_Normal.pdf