**Specifying Prior Probabilities in Bayesian Network by Maximum Likelihood Estimation method**

Loc Nguyen

University of Science, Ho Chi Minh city, Vietnam

Email: ng\_phloc@yahoo.com

### Abstract

Bayesian network provides the solid inference mechanism when convincing the hypothesis by collecting evidences. Bayesian network is instituted of two models such as qualitative model quantitative model. The qualitative model is its structure and the quantitative model is its parameters, namely conditional probability tables (CPT) whose entries are probabilities quantifying the dependences among variables in network. The quality of CPT depends on the initialized values of its entries. Such initial values are prior probabilities. Because the beta function provides some conveniences when specifying CPT (s), this function is used as the basic distribution in my method. The main problem of defining prior probabilities is how to estimate parameters in beta distribution. It is slightly unfortunate when the equations whose solutions are parameter estimators are differential equations and it is too difficult to solve them. By applying the maximum likelihood estimation (MLE) technique, I invent the simple equations so that differential equations are eliminated and it is much easier to estimate parameters in case that such parameters are positive integer numbers. Thus, I also propose the algorithm to find out the approximate solutions of these simple equations.

**Keywords**: Maximum likelihood estimation, beta distribution, beta likelihood estimation, gamma function, Bayesian network parameter.

### 1. Introduction

Bayesian network (BN) provides the solid inference mechanism when convincing the hypothesis by collecting evidences. BN is instituted of two models such as qualitative model and quantitative model. The qualitative model is its structure and the quantitative model is its parameters, namely conditional probability tables (CPT) whose entries are probabilities quantifying relationships among variables in network. The quality of CPT depends on the initialized values of its entries. Such initial values are prior probabilities. Because the beta function provides some conveniences when specifying CPT (s), this function is used as the basic distribution in this method. The main problem of defining prior probabilities is how to estimate parameters in beta distribution. It is slightly unfortunate when the equations whose solutions are parameter estimators are differential equations and it is too difficult to solve them. By applying the Maximum Likelihood Estimation (MLE) technique, a simple equation was constructed so that differential equations are eliminated and it becomes much easier to estimate parameters in case that such parameters are positive integer numbers (Nguyen L. , Specifying Prior Probabilities in Bayesian Network by Maximum Likelihood Estimation method, 2016). The algorithm to find out the approximate solutions of these simple equations is also proposed in this study. Recall that there are two ways to improve BN, parameter learning, which is also known as evolution of Bayesian overlay model mentioned in (Nguyen & Do, Combination of Bayesian Network and Overlay Model in User Modeling, 2009) and structure learning, which is based on dynamic Bayesian network described in (Nguyen L. , A New Algorithm for Modeling and Inferring User’s Knowledge by Using Dynamic Bayesian Network, 2014). The proposed algorithm specifying prior probabilities introduced here is essentially a parameter learning technique like evolution of Bayesian network mentioned in (Nguyen & Do, Evolution of parameters in Bayesian Overlay Model, 2009) but their difference is explained as follows:

* *Specifying prior probabilities* is to construct parameters (CPT (s)) of BN based on training data or data sample when BN has no CPT (s) yet.
* *Evolution of Bayesian network* is to improve or update CPT (s) when BN has already CPT (s), which means that specifying prior probabilities is always done before evolution of Bayesian network.

Recall that Bayesian network (BN) is the directed acyclic graph (DAG) constituted of a set of nodes representing random variables and a set of directed arcs representing relationships among nodes; please see (Neapolitan, 2003, p. 40) for more details about BN. The strengths of relationships are quantified by conditional probabilities. Each node owns a conditional probability table (CPT) that measures the impact of all its parents on it. Such CPT (s) are called the parameters of BN. Note that each entry in a CPT is a conditional probability. The problem which must be solved is how to initialize these parameters so as to be optimal. It means that we should specify prior probabilities.

Every node *X* in BN is a binary random variable. Each variable *X* is attached by a dummy variable *F* so that the probability density function (PDF) of such variable *F* represents CPT of *X*. The PDF of *F* conforms beta density function *β*(*F*; *a*, *b*) where *a* and *b* are two parameters. In other words, *F* has beta density function *β*(*F*; *a*, *b*) which is expressed by formula [1.1](#_Formula_III.1.49._Beta).

**Formula 1.1.** Beta density function *β*(*F*; *a*, *b*)

Where denotes gamma function. The expectation *E*(*F*) and the variance *Var*(*F*) of dummy variable *F* are:

The reason we choose beta density function as the probability distribution attached to every variable *X* in BN is that the prior probability of variable *X* is the expectation of *F* and it is very easy to compute this value according to formula [1.2](#_Formula_1.2._Probability):

**Formula 1.2.** Probability of variable *X* as expectation of beta variable *F*

Note that the equation implicates that dummy variable *F* is identified with its beta distribution *β*(*a*, *b*). We need to compute the posterior probability of variable *X* denoted as *P*(*X=*1*|*) where is the set of evidences in which the number of evidences having value 1 is *s* and the number of evidences having value 0 is *t*. Formula [1.3](#_Formula_III.1.51._Posterior) specifies posterior probability of *X* as conditional expectation of beta variable *F*.

**Formula 1.3.** Posterior probability of variable *X* as conditional expectation of beta variable *F*

Where *N=a+b* and *M=s+t*.

Note that the equation implicates that dummy variable *F* is identified with its beta distribution *β*(*a*, *b*).

It is recognized that beta distribution provides us convenience when specifying CPT (s) in BN. It is essential to count the number of evidences so as to compute the posterior probabilities. However, the quality of CPT is also dependent on the prior probability and so; the considerable problem is involved in how to estimate two parameters of beta distribution *a* and *b* because the prior probability is derived from them, .

Section [2](#_III.1.5.1._Maximum_likelihood) discusses some basic concepts of maximum likelihood estimation (MLE) technique. Section [3](#_III.1.5.2._Beta_likelihood) considers applying MLE into beta distribution (beta density function). Section [4](#_III.1.5.3._Algorithm_to) – the main sub-section describes the proposed algorithm to estimate two parameters *a* and *b* of beta distribution. Section [4](#_III.1.5.3._Algorithm_to) also proposes the simple equations whose solutions are estimates of positive parameters *a* and *b*. Section [5](#_III.1.5.4._An_example) illustrates the proposed algorithm by example. Section [6](#_III.1.5.6._Evaluation) is the conclusion.

### 2. Maximum likelihood estimation

Let and *X* be the hypothesis and observation variable, respectively. Suppose *x*1, *x*2,…, *xn* are instances of variable *X* in training data and they are observed independently. In study of statistics, training data is called sample which is constituted of these observations or evidences *xi* (s) and such *xi* (s) are considered as independent and identically distributed (i.i.d) random variables. This means that *xi* (s) and *X* have the same probability distribution or the same probability density function (PDF). According multiplication rule in probability theory, the likelihood function *L*() is the joint probability which is the product of condition probabilities of instances *xi*, given hypothesis variable (Lynch, 2007, p. 36). Formula [2.1](#_Formula_III.1.52._Likelihood) expresses the likelihood function *L*() with regard to variable .

**Formula 2.1.** Likelihood function

Where is the conditional probability of instance *xi* given the hypothesis . Suppose = {*θ*1, *θ*2,…, *θk*} is the vector of parameters specifying the distribution of *X* (density function of *X*) denoted *f*. It is required to estimate the parameter vector and its standard deviation in distribution *f* so that the likelihood function takes the maximum value. Thus, this method is called maximum likelihood estimation (MLE). The parameter vector that maximizes likelihood function is called *optimal parameter vector* or *parameter vector estimator* denoted , as shown in formula [2.2](#_Formula_III.1.51._Optimal). If was evaluated, it can be considered *parameter vector estimate*. It is possible to use terms such as “optimal parameter vector”, “parameter vector estimator”, and “parameter vector estimate” exchangeably. We can remove the word “vector” inside these terms if we do not focus on the fact that is a vector. In other words, can be called as optimal parameter, parameter estimator, and parameter estimate.

**Formula 2.2.** Optimal parameter vector

Because it is too difficult to work with the likelihood function in the form of product of condition probabilities, it is necessary to take logarithm of *L*() so as to transform the likelihood function from form of repeated multiplication, as shown in formula [2.1](#_Formula_III.1.52._Likelihood), into form of repeated addition, as shown in formula [2.3](#_Formula_III.1.52._Log-likelihood) (Lynch, 2007, p. 38). The natural logarithm of *L*(), which is called log-likelihood, is denoted *LnL*(), as shown in formula [2.3](#_Formula_III.1.52._Log-likelihood).

**Formula 2.3.** Log-likelihood function and optimal parameter vector

Where *ln*(.) denotes natural logarithm function.

The essence of maximizing the likelihood function is to find the peak of the curve of *LnL*() (Lynch, 2007, p. 38). This can be done by setting the first-order partial derivative of *LnL*() with respect to each parameter *θi* to 0 and solving this equation to find out parameter *θi*. The number of equations corresponds with the number of parameters. If all parameters are found, in other words, the optimal parameter vector  *=*  is defined then, the optimal distribution is known clearly. Each is also called a *parameter estimator*; on the other hand can be considered *parameter estimate* or *optimal parameter* if it is evaluated as numeric value. It is possible to use terms such as “optimal parameter”, “parameter estimator”, and “parameter estimate” exchangeably.

The accuracy of parameter estimator is measured by its standard error (Montgomery & Runger, 2003, p. 225) and thus; another important issue is how to determine the standard error in distribution *f* when we have already computed all parameters and standard error is standard deviation of parameter estimator. It is very fortunate when the second-order derivative of the log-likelihood function denoted can be computed and it is used to determine the variances of parameters. If distribution *f* has only one parameter, the second-order derivative is scalar, otherwise it is a so-called Hessian matrix. The negative expectation of Hessian matrix is called the *information matrix* which in turn is inverted so as to construct *co-variance matrix* denoted *Var*() (Lynch, 2007, p. 40). Formula [2.4](#_Formula_III.1.55._Co-variance) specifies the co-variance matrix of parameter vector .

**Formula 2.4.** Co-variance matrix of parameter vector

Elements on co-variance matrix diagonal are variances of the parameters and the square root of each variance is a standard error. Exactly, is a so-called Cramer-Rao lower bound of co-variance matrix. However, is equal to co-variance matrix if is unbiased estimator (Zivot, 2009, p. 11). Please read (Lynch, 2007, pp. 35-43) and (Zivot, 2009) for more details about MLE.

Next section [3](#_III.1.5.2._Beta_likelihood) – “Beta likelihood estimation” discusses how to apply MLE into beta distribution (beta density function).

### 3. Beta likelihood estimation

As discussed, each variable *X* within BN is attached by a dummy variable *F* and variable *F*, in turn, has beta distribution *β*(*F*; *a*, *b*) specified in the formula [1.1](#_Formula_III.1.49._Beta). For convenience, the beta density function specified by previous formula [1.1](#_Formula_III.1.49._Beta) is re-written as formula [3.1](#_Formula_III.1.55._Beta).

**Formula 3.1.** Beta density function (beta distribution) *β*(*F*; *a*, *b*)

Where Γ(.) denotes gamma function as follows:

Note that *e*(.) and *exp*(.) denote exponent function and *e*2.71828 is Euler’s number.

Beta density function is based on gamma function and there is another so-called *digamma function* is also defined via gamma function.Formula[3.2](#_Formula_III.1.57._Definition)is definition of digamma function *ψ*(*x*). We will know later that beta density function is also relevant to digamma function.

**Formula 3.2.** Definition of digamma function

Note that *ln*(.) denotes natural logarithm function. According to formula[3.2](#_Formula_III.1.57._Definition), digamma function is the derivative of natural logarithm of gamma function.

The integral form of digamma function is specified by formula [3.3](#_Formula_III.1.58._Integral) (Medina & Moll, 2009, p. 114):

**Formula 3.3.** Integral form of digamma function

Let *ψ*1(*x*) be the first-order of digamma function, we have:

(because function is continuous and differentiable in open interval (0, +∞) with regard to variable *x*)

We also have:

Function *ψ*1(*x*) is also called *trigamma* function; please refer to documents (Weisstein, Polygamma Function), (Wikipedia, Polygamma function, 2014), and (Weisstein, Trigamma Function) for more details about trigamma function. Briefly, formula [3.4](#_Formula_III.1.65._The) expresses trigamma function.

**Formula 3.4.** Trigamma function

The beta function (Wikipedia, Beta function, 2014) denoted *B*(*x, y*) is a special function defined as below:

**Formula 3.5.** Beta function *B*(*x*, *y*)

Please distinguish beta density function *β*(*X*; *a*, *b*) specified in formulas [3.1](#_Formula_III.1.55._Beta) and [1.1](#_Formula_III.1.49._Beta) known as probability density function (PDF) from beta function *B*(*x*, *y*) specified by formula [3.5](#_Formula_III.1.64._Beta).

The first-order partial derivative of *B*(*x, y*) is determined as follows:

Due to digamma function , we have formula [3.6](#_Formula_III.1.57._First-order) to specify the first-order partial derivative of beta function:

**Formula 3.6.** First-order partial derivative of beta function *B*(*x*, *y*)

The digamma function is always determined by formulas [3.2](#_Formula_III.1.57._Definition) and [3.3](#_Formula_III.1.58._Integral). Substituting beta function *B*(*x*, *y*) specified formula [3.5](#_Formula_III.1.64._Beta) into formula [3.1](#_Formula_III.1.55._Beta), the beta density function is re-written:

**Formula 3.7.** Beta density function with regard to beta function

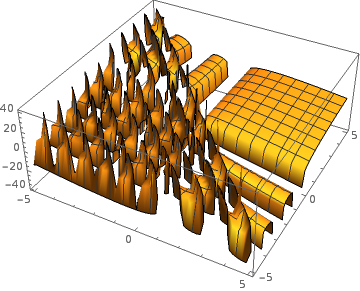
Now we specify the likelihood function of beta distribution by applying formula [3.7](#_Formula_III.5.2.11._Beta) as below:

Where *fi* is physical frequency of a specified event in sample with note that 0 < *fi* < 1 and so this likelihood can use *n* samples. Taking the logarithm of *L*(*a*, *b*), we have the log-likelihood function for beta distribution as follows (Wikipedia, Beta distribution, 2018):

**Formula 3.8.** Log-likelihood function of beta density function (beta distribution)

Where *ln*(.) denotes natural logarithm function.

Figure [3.1](#_Figure_III.5.2.1._Log-likelihood) (Wolfram) shows the graph of log-likelihood function specified by formula [3.8](#_Formula_III.1.67._Log-likelihood) with regard to variables *a* and *b* given *f*1=0.1 and *f*2=0.2.



**Figure 3.1.** Log-likelihood function with regard to variables *a* and *b*

Please pay attention to formula [3.8](#_Formula_III.1.67._Log-likelihood) because formula [3.8](#_Formula_III.1.67._Log-likelihood) is specific case of formula [2.3](#_Formula_III.1.52._Log-likelihood) mentioned in previous section [2](#_III.1.5.1._Maximum_likelihood); thus, MLE is applied into beta distribution.

Two parameters *a* and *b* must be determined so that they maximize the log-likelihood function. Thus, by taking two first-order partial derivatives of log-likelihood function specified in formula [3.8](#_Formula_III.1.67._Log-likelihood) corresponding to two parameters and by applying formula [3.6](#_Formula_III.1.65._First-order), we have:

**Formula 3.9.** First-order partial derivative of log-likelihood function of beta density function with regard to parameter *a*

**Formula 3.10.** First-order partial derivative of log-likelihood function of beta density function with regard to parameter *b*

Where is digamma function specified by formulas [3.2](#_Formula_III.1.57._Definition) and [3.3](#_Formula_III.1.58._Integral). Note that notation denotes first-order partial derivative of multi-variable function *g* with regard to variable *x*.

Please pay attention to formulas [3.9](#_Formula_III.1.60._The) and [3.10](#_Formula_III.1.61._The) for determining two first-order partial derivatives of log-likelihood function of beta distribution. By setting such two partial derivatives equal 0 so as to find out two parameters *a* and *b*, we have a set of equations whose two solutions are the values of *a* and *b*:

**Formula 3.11.** The set of differential equations for estimating parameters *a* and *b*

Formula [3.11](#_Formula_III.1.63._Two) shows the set of differential equations for estimating parameters *a* and *b*. In the next section [4](#_III.1.5.3._Algorithm_to), the simple form of such equations is invented when parameters *a* and *b* are positive integer numbers. Furthermore, the algorithm to find out the approximate solutions is derived.

Before going into the main section [4](#_III.1.5.3._Algorithm_to), it is necessary to glance over the co-variance matrix *Var*(*a*, *b*) of parameters of beta density function mentioned in previous section [2](#_III.1.5.1._Maximum_likelihood) although the co-variance matrix is not important subject in the research. Let *H*(*a*, *b*) be the second-order partial derivative matrix called Hessian matrix, we have:

Note, the bracket (.) denotes matrix.

Basing on formulas [3.9](#_Formula_III.1.60._The) and [3.10](#_Formula_III.1.61._The), we can determine four second-order partial derivatives of log-likelihood function as follows:

Where *ψ*1(.) denotes trigamma function specified by formula [3.4](#_Formula_III.1.65._The). According to formula [2.4](#_Formula_III.1.55._Co-variance), the co-variance matrix *Var*(*a*, *b*) is the inversion of negative expectation of Hessian matrix. Please read the book “Linear Algebra” by author Viet-Hung Huu Nguyen (Nguyen V. H., 1999, p. 134) and the book “Linear Algebra and Its Applications” by author Lay (Lay, 2012, pp. 102-109) for more details of how to take inversion of a given matrix. We have:

(Because trigamma functions *ψ*1(*a*), *ψ*1(*b*), and *ψ*1(*a+b*) are only dependent on parameters *a* and *b*, the expectation of *H*(*a*, *b*) is merely *H*(*a*, *b*))

Briefly, formula [3.12](#_Formula_III.1.67._Co-variance) specifies the co-variance matrix of parameters of beta density function as follows:

**Formula 3.12.** Co-variance matrix of parameters of beta density function

Where *ψ*1(*.*) denotes trigamma function and,

The roots of diagonal elements are the standard deviations (standard errors) of parameter estimates. Let and be the standard errors of optimal parameters and where and are solutions of equations specified by formula [3.11](#_Formula_III.1.63._Two), we have:

**Formula 3.13.** Standard errors of parameter estimates of beta distribution

Where *ψ*1(*.*) denotes the trigamma function and,

Formula [3.13](#_Formula_III.1.72._Standard) specifying standard errors of parameter estimates ends up this section [3](#_III.1.5.2._Beta_likelihood) mentioning applying MLE technique into beta distribution. Now the next section [4](#_III.1.5.3._Algorithm_to) will mention the proposed approach to solve the set of differential equations specified in formula [3.11](#_Formula_III.1.63._Two).

### 4. Algorithm to solve the equations whose solutions are parameter estimators

As specified by formula [3.11](#_Formula_III.1.63._Two), the parameter estimators and are solutions of two equations:

Obviously, these equations are differential functions whose solutions are families of functions. Because it is too difficult to solve such equations, a simple form of them is discovered with parameters *a* and *b* being positive integer numbers and hence, differential functions are eliminated from the simple form.

Suppose that parameters *a* and *b* are positive integer numbers. Expanding the expression *ψ*(*a*) *– ψ*(*a+b*), we have:

Due to,

Expending the polynomials and with note that *a* and *b* are positive integer numbers, we also have

Where is the combination taken *k* of a + *b* – 1 elements and is the combination taken *k* of *a* – 1 elements,

Hence, we have

Without loss of generality, suppose *a* ≥ 2, we have:

In the similar way, given *b* ≥ 2 we have:

The equations whose solutions are parameter estimators becomes two following equations:

**Formula 4.1.** Two simplest equations for estimating positive integer parameters *a* and *b*

Where,

Obviously, formula [4.1](#_Formula_III.1.73._Two) is the simplest form from which all differential functions are removed. However, the number of solutions of equations in formula [4.1](#_Formula_III.1.73._Two) is large and it is very difficult to find out them. Suppose there is the restriction:

*“The range of variables a and b is from* 1 *to n where n is the whole positive number and not greater than the number of evidences in training data”.*

I propose the iterative algorithm that each pair values (*ai, bi*) which are values of variables *a* and *b* are fed to *G*1, *G*2 at each iteration. Two biases Δ1*=G*1(*ai*, *bi*)–*L*1 and Δ2=*G*2(*ai*, *bi*)–*L*2 are computed. The normal bias is the root of sum of the second power Δ1 and the second power of Δ2 and so we have Δ*=.* The pair (, ) whose normal bias Δ is minimum are chosen as the parameter estimators. The algorithm is described in table [4.1](#_Table_III.1.25._Iterative) as below:

|  |
| --- |
| *min*Δ = +∞  = *=* 1 (uniform distribution )  For *a=*1 to *n* do  For *b=*1 to *n* do  Δ1*=G*1(*a*, *b*)*–L*1  Δ2*=G*2(*a*, *b*)–*L*2  Δ*=*  IfΔ *< min*Δthen  *min*Δ=Δ  *=a*  *=b*  End If  End For *a*  End For *b*  ( and are optimal parameters) |

###### **Table 4.1.** Iterative algorithm solving simplest equations specified by formula [4.1](#_Formula_III.1.63._Two) for estimating parameters *a* and *b*

Where *min*Δ denotes minimum bias.

Note that the power and combination functions occurring in *G*1 and *G*2 become unexpectedly huge when the range of *a* and *b* is wide; so the upper bound *n* should not be large. With respect to beta distribution, the probability of variable in Bayesian network is the expectation of beta distribution, namely, *P*(*X*) *= E*(*β*(*a, b*)) *=* ; please see formula [1.2](#_Formula_III.1.50._Probability) for knowing probability of *X* as expectation of beta distribution. The ratio is not so dependent on the amplitude of *a* or *b*; for example, the ratio whose parameters *a* and *b* equal 5 and 7, respectively is the same to the ratio whose parameters *a* and *b* equal 10 and 14, respectively. That is why we do not need to define the range of *a* and *b* to be so wide.

The next section [5](#_III.1.5.4._An_example) describes an example illustrating proposed algorithm in table [4.1](#_Table_III.1.25._Iterative).

### 5. An example of how to specify prior probabilities

Suppose there is the BN having two variables *X*1, *X*2 and one arc which links them together. Variables *X*1 and *X*2 obey beta distribution. We need to specify the prior CPT (s), namely the prior conditional probabilities *P*(*X*1=1), *P*(*X*2=1*|X*1=1) and *P*(*X*2=1*|X*1=0). Figure [5.1](#_Figure_III.1.28._Bayesian) depicts the example of BN including such variables *X*1 and *X*2 without CPT (s).

*P*(*X*1=1) *P*(*X*1=0)

? ?

*X*1 *P*(*X*2=1)

1. ?

0 ?

**Figure 5.1.** Bayesian network without CPT (s)

In figure [5.1](#_Figure_III.1.28._Bayesian), question marks (?) indicate undefined conditional probabilities.

Let *β*1(*a*1,*b*1), *β*2(*a*2,*b*2), *β*3(*a*3,*b*3) be beta distributions of conditional probabilities *P*(*X*1=1), *P*(*X*2=1*|X*1=1) and *P*(*X*2=1*|X*1=0). Applying formula [1.2](#_Formula_III.1.50._Probability) for specifying probability of *X* as expectation of beta distribution, we have:

It is necessary to determine three parameter pairs (*a*1, *b*1), (*a*2, *b*2)and(*a*3, *b*3) of three beta distributions *β*1, *β*2and *β*3, respectively. Suppose we perform 5 trials of a random process. Outcome of the *ith* trial denoted *D*(*i*)is considered as an evidence in which *X*1 and *X*2 obtain value 0 or 1. So we have one sample  *=* (*D*(1), *D*(2), *D*(2), *D*(3), *D*(4), *D*(5)). Table [5.1](#_Table_III.1.26._The) shows these five evidences.

|  |  |  |
| --- | --- | --- |
|  | *X*1 | *X*2 |
| *D*(1) | *X*1= 1 | *X*2 = 1 |
| *D*(2) | *X*1= 1 | *X*2 = 1 |
| *D*(3) | *X*1= 1 | *X*2 = 0 |
| *D*(4) | *X*1= 0 | *X*2 = 1 |
| *D*(5) | *X*1= 0 | *X*2 = 0 |

###### **Table 5.1.** The evidences corresponding to 5 trials

According to the proposed algorithm described in table [4.1](#_Table_III.1.25._Iterative), let *Lij*, *Gij*,Δ*ij*,Δ*i* be the values of *Lj*, *Gj*,Δ*j*,Δ with respect to *βi* where , , and *n* = 1. Because there is only one sample (*n*=1), we have:

* *L*11=*ln*(*fi*) and *L*12=*ln*(1–*fi*) where *fi*is the frequency of event *X*1=1.
* *L*21=*ln*(*fi*) and *L*22=*ln*(1–*fi*) where *fi*is the frequency of event *X*2=1 given *X*1=1.
* *L*31=*ln*(*fi*) and *L*32=*ln*(1–*fi*) where *fi*is the frequency of event *X*2=1 given *X*1=0.
* *G*11(*a*1, *b*1) = and *G*12(*a*1, *b*1)= .
* *G*21(*a*2, *b*2) = and *G*22(*a*2, *b*2)= .
* *G*31(*a*3, *b*3) = and *G*32(*a*3, *b*3)= .
* Δ11=*G*11 – *L*11, Δ12=*G*12 – *L*12 and Δ1=
* Δ21=*G*21 – *L*21, Δ22=*G*22 – *L*22 and Δ2=
* Δ31=*G*31 – *L*31, Δ32=*G*32 – *L*32 and Δ3=

From above evidences shown in table [5.1](#_Table_III.1.26._The), it is easy to compute *Lij*. For example, because there are 3 evidences *D*(1), *D*(2), *D*(2), *D*(3) among 5 evidences in which *X*1=1, we have *L*11 = *ln*(3/5) ≈ –0.51 and *L*12 = *ln*(1–3/5) ≈ –0.92. In the similar way, all *Lij* are determined. Table [5.2](#_Table_III.1.27._The) shows values of *Lij* corresponding to beta density functions *β*1, *β*2, and *β*3.

|  |  |  |
| --- | --- | --- |
| *β*1 | *β*2 | *β*3 |
| *L*11 = –0.51 | *L*21 = –0.41 | *L*31 = –0.69 |
| *L*12 = –0.92 | *L*22 = –1.10 | *L*32 = –0.69 |

###### **Table 5.2.** The values of *Lij* corresponding to beta density function *β*1, *β*2, and *β*3

Suppose the range of all parameters is from 1 to 4. By applying the proposed algorithm described in table [4.1](#_Table_III.1.25._Iterative), it is easy to compute the normal biases. For example, given *a*1 = 1 and *b*1 = 1, we have:

Followingtable [5.3](#_Table_III.1.28._The) shows normal biases of all possible values of (*a*1*, b*1).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***a*1** | ***b*1** | ***L*11** | ***L*12** | ***G*11** | ***G*12** | **Δ11** | **Δ12** | **Δ1** |
| 2.0 | 2.0 | –0.51 | –0.92 | –0.83 | –0.83 | –0.32 | 0.08 | 0.33 |
| 2.0 | 3.0 | –0.51 | –0.92 | –1.08 | –0.58 | –0.57 | 0.33 | 0.66 |
| 2.0 | 4.0 | –0.51 | –0.92 | –1.28 | –0.45 | –0.77 | 0.47 | 0.90 |
| 3.0 | 2.0 | –0.51 | –0.92 | –0.58 | –1.08 | –0.07 | –0.17 | 0.18 |
| 3.0 | 3.0 | –0.51 | –0.92 | –0.78 | –0.78 | –0.27 | 0.13 | 0.30 |
| 3.0 | 4.0 | –0.51 | –0.92 | –0.95 | –0.62 | –0.44 | 0.30 | 0.53 |
| 4.0 | 2.0 | –0.51 | –0.92 | –0.45 | –1.28 | 0.06 | –0.37 | 0.37 |
| **4.0** | **3.0** | **–0.51** | **–0.92** | **–0.62** | **–0.95** | **–0.11** | **–0.03** | **0.11** |
| 4.0 | 4.0 | –0.51 | –0.92 | –0.76 | –0.76 | –0.25 | 0.16 | 0.29 |

###### **Table 5.3.** The normal biases of (*a*1, *b*1) with respect to *β*1

The normal biases of all possible values of (*a*2, *b*2) with respect to *β*2 are shown in following table [5.4](#_Table_III.1.29._The).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***a*2** | ***b*2** | ***L*21** | ***L*22** | ***G*21** | ***G*22** | **Δ21** | **Δ22** | **Δ2** |
| 2.0 | 2.0 | –0.41 | –1.1 | –0.83 | –0.83 | –0.43 | 0.27 | 0.50 |
| 2.0 | 3.0 | –0.41 | –1.1 | –1.08 | –0.58 | –0.68 | 0.52 | 0.85 |
| 2.0 | 4.0 | –0.41 | –1.1 | –1.28 | –0.45 | –0.88 | 0.65 | 1.09 |
| **3.0** | **2.0** | **–0.41** | **–1.1** | **–0.58** | **–1.08** | **–0.18** | **0.02** | **0.18** |
| 3.0 | 3.0 | –0.41 | –1.1 | –0.78 | –0.78 | –0.38 | 0.32 | 0.49 |
| 3.0 | 4.0 | –0.41 | –1.1 | –0.95 | –0.62 | –0.54 | 0.48 | 0.73 |
| 4.0 | 2.0 | –0.41 | –1.1 | –0.45 | –1.28 | –0.04 | –0.18 | 0.19 |
| 4.0 | 3.0 | –0.41 | –1.1 | –0.62 | –0.95 | –0.21 | 0.15 | 0.26 |
| 4.0 | 4.0 | –0.41 | –1.1 | –0.76 | –0.76 | –0.35 | 0.34 | 0.49 |

###### **Table 5.4.** The normal biases of (*a*2, *b*2) with respect to *β*2

The normal biases of all possible values of (*a*3, *b*3) with respect to *β*3 are shown in following table [5.5](#_Table_III.1.30._The).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***a*3** | ***b*3** | ***L*31** | ***L*32** | ***G*31** | ***G*32** | **Δ31** | **Δ32** | **Δ3** |
| 2.0 | 2.0 | –0.69 | –0.69 | –0.83 | –0.83 | –0.14 | –0.14 | 0.20 |
| 2.0 | 3.0 | –0.69 | –0.69 | –1.08 | –0.58 | –0.39 | 0.11 | 0.41 |
| 2.0 | 4.0 | –0.69 | –0.69 | –1.28 | –0.45 | –0.59 | 0.24 | 0.64 |
| 3.0 | 2.0 | –0.69 | –0.69 | –0.58 | –1.08 | 0.11 | –0.39 | 0.41 |
| 3.0 | 3.0 | –0.69 | –0.69 | –0.78 | –0.78 | –0.09 | –0.09 | 0.13 |
| 3.0 | 4.0 | –0.69 | –0.69 | –0.95 | –0.62 | –0.26 | 0.08 | 0.27 |
| 4.0 | 2.0 | –0.69 | –0.69 | –0.45 | –1.28 | 0.24 | –0.59 | 0.64 |
| 4.0 | 3.0 | –0.69 | –0.69 | –0.62 | –0.95 | 0.08 | –0.26 | 0.27 |
| **4.0** | **4.0** | **–0.69** | **–0.69** | **–0.76** | **–0.76** | **–0.07** | **–0.07** | **0.09** |

###### **Table 5.5.** The normal biases of (*a*3, *b*3) with respect to *β*3

From above tables [5.3](#_Table_III.1.28._The), [5.4](#_Table_III.1.29._The), and [5.5](#_Table_III.1.30._The), we recognize that when **(*a*1,*b*1)=(4,3), (*a*2,*b*2)=(3,2)** and **(*a*3,*b*3)=(4,4)**, the normal biases of distributions *β*1, *β*2 and *β*3, respectively become minimum. So the parameter estimators , and corresponding to distributions *β*1, *β*2 and *β*3 are (4,3), (3,2) and (4,4), respectively. So the prior conditional probabilities *P*(*X*1=1), *P*(*X*2=1*|X*1=1), and *P*(*X*2=1*|X*1=0) are determined:

When these prior probabilities were calculated, the Bayesian network is totally determined with full of prior CPT (s) as in figure [5.2](#_Figure_III.1.25._Bayesian).

*P*(*X*1=1) *P*(*X*1=0)

**0.57** 0.43

*X*1 *P*(*X*2=1)

1. **0.60**

0 **0.50**

**Figure 5.2.** Bayesian network with full of prior CPT (s)

The figure [5.2](#_Figure_III.1.29._Bayesian) shows the ultimate result of the interesting algorithm mentioned mainly in section [4](#_III.1.5.3._Algorithm_to), which is used for specifying prior CPT (s) of Bayesian network.

Let , , , , , and be standard errors of , , , , , and . By applying formula [3.13](#_Formula_III.1.72._Standard), it is easy to determine these standard errors as follows:

The errors and are minimum, which implies that and are best estimates.

In general, the iterative algorithm takes advantages of simple equations whose solutions are estimates of positive parameter *a* and *b*. Section [6](#_III.1.5.6._Evaluation) is the conclusion.

### 6. Conclusion

The basic idea of MLE is to solve the equation formed by setting the first-order derivation of log-likelihood function equal 0. I apply MLE into beta distribution so as to determine the equations for computing two parameters of beta distribution. Because beta distribution is more complex to estimate its parameters than other distributions like binominal distribution and normal distribution, I invent the simple form of MLE equations in case that parameters are positive integer numbers. Moreover, I propose the algorithm so as to find out approximate solutions of these equations. This is iterative algorithm in which a number of parameters are surveyed and the parameter whose bias with respect to the actual solution is minimum is the approximate solution. It is impossible to find out the precise solutions but it is easy and feasible to implement our algorithm as computer program. Although the simple form of MLE equations and the iterative algorithm are my two contributions but the most important is the invention of simple MLE equations. The ideology behind the simple equations is that we focus on finding out discrete parameters instead of making effort to solve differential equations and moreover, the proposed iterative algorithm is only appropriate to such ideology. I think that the invention is worthy to publish worldwide.

### References

Lay, D. C. (2012). *Linear Algebra and its applications* (4th Edition ed.). (D. Lynch, Ed.) Boston, Maryland, USA: Pearson Education.

Lynch, S. M. (2007). *Introduction to Applied Bayesian Statistics and Estimation for Social Scientists.* Springer Berlin Heidelberg NewYork.

Medina, L. A., & Moll, V. H. (2009). The integrals in Gradshteyn and Ryzhik. Part 10: The digamma function. *Series A: Mathematical Sciences, 17*(2009), 45-66.

Montgomery, D. C., & Runger, G. C. (2003). *Applied Statistics and Probability for Engineers* (3rd Edition ed.). New York, NY, USA: John Wiley & Sons, Inc.

Neapolitan, R. E. (2003). *Learning Bayesian Networks.* Upper Saddle River, New Jersey, USA: Prentice Hall.

Nguyen, L. (2014, May). A New Algorithm for Modeling and Inferring User’s Knowledge by Using Dynamic Bayesian Network. (M. Z. Raqab, Ed.) *Statistics Research Letters (SRL), 3*(2). Retrieved from http://www.srl-journal.org/paperInfo.aspx?ID=6933

Nguyen, L. (2016, February). Specifying Prior Probabilities in Bayesian Network by Maximum Likelihood Estimation method. (B. N. Buszewski, Ed.) *Sylwan Journal, 160*(2), 281-298.

Nguyen, L., & Do, P. (2009). Combination of Bayesian Network and Overlay Model in User Modeling. (M. E. Auer, Ed.) *International Journal of Emerging Technologies in Learning (iJET), 4*(4), 41-45. doi:10.3991/ijet.v4i4.684

Nguyen, L., & Do, P. (2009). Evolution of parameters in Bayesian Overlay Model. In H. R. Arabnia, D. d. Fuente, & J. A. Olivas (Ed.), *Proceedings of The 2009 International Conference on Artificial Intelligence (IC-AI’09), The 2009 World Congress in Computer Science, Computer Engineering, and Applied Computing (WORLDCOMP’09)* (pp. 324-329). Monte Carlo Resort, Las Vegas, Nevada, USA: CSREA Press USA. Retrieved from http://www.researchgate.net/publication/220834553\_Evolution\_of\_Parameters\_in\_Bayesian\_Overlay\_Model

Nguyen, V. H. (1999). *Linear Algebra.* Hanoi, Vietnam: Hanoi National University Publishing House.

Weisstein, E. W. (n.d.). *Polygamma Function*. (Wolfram Research) Retrieved November 13, 2014, from MathWorld--A Wolfram Web Resource: http://mathworld.wolfram.com/PolygammaFunction.html

Weisstein, E. W. (n.d.). *Trigamma Function*. (Wolfram Research) Retrieved November 13, 2014, from MathWorld--A Wolfram Web Resource: http://mathworld.wolfram.com/TrigammaFunction.html

Wikipedia. (2014, October 16). *Beta function*. (Wikimedia Foundation) Retrieved November 9, 2014, from Wikipedia website: http://en.wikipedia.org/wiki/Beta\_function

Wikipedia. (2014, October 12). *Polygamma function*. (Wikimedia Foundation) Retrieved November 10, 2014, from Wikipedia website: https://en.wikipedia.org/wiki/Polygamma\_function

Wikipedia. (2018, March 30). *Beta distribution*. (Wikimedia Foundation) Retrieved 2008, from Wikipedia website: https://en.wikipedia.org/wiki/Beta\_distribution

Wolfram. (n.d.). Mathematica. *Wolfram Mathematica, 10(4)*. Wolfram Research. Retrieved March 2016, from http://www.wolfram.com/mathematica

Zivot, E. (2009). *Maximum Likelihood Estimation.* Lecture Notes on course "Econometric Theory I: Estimation and Inference (first quarter, second year PhD)", University of Washington, Seattle, Washington, USA.