BAYESIAN NORMALITY RELAXATION USING NEO-NORMAL DISTRIBUTION AND ITS IMPLEMENTATION FOR DATA ANALYSIS VIA WINBUGS

Nur Irwian, Dedy Dwi Prasetyo, and Restika Shary
Statistics Department, Faculty of Mathematics and Natural Sciences, Institut Teknologi Sepuluh Nopember, Surabaya - Indonesia

Abstract. Normality relaxation seems nasty to be done in data analysis, but every data modelling normal assumption is not always can be fulfilled. Bayesian approaches are granting broadly to do these cases. This paper describes how we use neo-normal densities as a tool of Bayesian data analysis in WinBUGS. Some neo-normal distributions founded by researchers would be considered in this paper. The works of these additional tools will be demonstrated for identifying densities and for estimating linear model of some simulation and real data. The convergence of MCMC run in WinBUGS would guarantee the precision and accuracy of the estimation. Finally, by employing this normality relaxation using Bayesian approaches, the estimation results shows more realistic than other methods of estimation in WinBUGS.

Keywords: Neo-normal distribution; Bayesian; MCMC; Linear model.

Introduction
Limited number of data in the statistical modelling analysis makes the ordinary Normal error assumption, IIDN(0,σ^2), frequently not appropriate to be used. One way to solve this problem, people frequently apply Bayesian methods by increasing some prior knowledge related to the parameters of the model. It is due to analytical non-normal assumptions are often far from the reality and non-normal modelling cannot be avoided. Some works as Exponential Power distribution by Box and Tiao (1973), Stable distribution by Buckle (1997), Skewed Student distribution by Fernandez and Steel (1998), and neo-normal distribution by Irie (2000) have succeeded to overcome these problems in both cases of density and linear models estimation as a normality relaxation in data analysis and modelling. Indeed, an open source R-program has implemented skewed Student on it. Motivated by those works, this paper will demonstrate Bayesian neo-modelling using WinBUGS that has been added by a new flexible and adaptive neo-normal distribution.

WinBUGS release 1.4 is a Bayesian tools which can be found by free in internet (Lunn et al, 2000). WinBUGS is the current, windows-based, version of the BUGS software described in Spiegelhalter et al. (1996). WinBUGS stands for Windows Bayesian inference Using Gibbs Sampling. Its object-oriented form couple with its 'point-and-click' environment makes more user-friendly and easier to be used. Lunn and Jackson (2004) give an interesting help for creative WinBUGS user in developing own distribution. This guidance very useful for user who has specific problems with particular particular data which does not supported by WinBUGS utilities.

The structure of Bayesian modeling which shows hierarchical construction has been prepared com by a directed acyclic graph using Doodle tools. New innovation of Bayesian modeling such as neo-normality in the form of linear model error modeling would be easily supported by this. The primary method of Markov Chain Monte Carlo (MCMC) as the state-of-the-art statistical tool which also as a wide class of Bayesian full probability models are more accessible for analysing and comparing the model is then conducted using various simulation methods known as Gibbs...
Finally, some advantages of the use of normality relaxation identification to simulated skewed data production during the year of 2000 in East Java cited from BPS will be shown in this paper.

Some Normality Relaxation Inspections have been investigated

Some distributions which are classified as normality relaxation have been created. They have a special parameter that can be used to monitor the data or distribution of error of linear model is categorized non-normal or not. Exponential Power distribution used by Box and Tiao (1973) has parameter \( \beta \) to the neo-normality. The domain of \( \beta \) is \(-1 < \beta < 1\). When \( \beta = 0 \), then the data will be normally distributed and therefore the analysis can be done using classic methods. While \( \beta \neq 0 \), analysis using relaxation-normality approaches is needed. Because, if \( \beta = -1 \), the distribution of error will be almost uniformly distributed, fat-tails, and platikurtic. On the other hand, it will be detected as double exponential distribution that has fat-tails and leptokurtic, when \( \beta = 1 \). Exponential power distribution can only modelled data asymmetrical form, even though the data is actually more skew.

The second distribution is Skewed Student and skewed normal distribution established by Fernandez and Steel (1998) using parameter \( \gamma \) to identify the neo-normality. This parameter changes in \((0, \infty)\). This distribution can be able to catch that the data is symmetrical or skew. As \( \gamma = 0 \) in exponential distribution, most preferable condition in this distribution is when \( \gamma = 1 \), which reports the data is normally or standardly distributed. When \( \gamma < 1 \), the distribution reports that the data skew to right and conversely for \( \gamma > 1 \). The third distribution is MSNBurr and MSTDurr distribution as a neo-normal representation by Iriawan (2000) using parameter \( \alpha \) to recognize the neo-normality. The domain of \( \alpha \) is as \( \gamma \) in skew normal student, \((0, \infty)\). But, the way to identify the skewness is different. The data is identified to be skewed to the left when \( \alpha < 1 \) and conversely for skewed to the right.

As has been shown in Box and Tiao (1973), Fernandez and Steel (1998), and even in Iriawan (2000), these distributions have already demonstrated for neo-normal density estimation and also considered the pattern of linear model error. All of distribution above employ Bayesian approaches to implement estimating parameters, but do not take benefit from powerfulness of WinBUGS.

Estimating Neo-Normal Using Bayesian Procedures in WinBUGS

Simplest procedure in estimating parameter of model is always as a target of statistical data modeling analysis. When the data is supposed to be MSNBurr or MSTDurr distributed, with cumulative distribution function (CDF) and probability density function (pdf) of these distributions are as seen in (1) and (2) (Iriawan, 2000), then maximum likelihood methods looks very complicated to be used.

\[
F(x) = 1 + \frac{1}{\alpha} \exp \left( -k \frac{x - \mu}{\phi} \right) ^{- (\alpha + 1)}
\]

\[
f(x | \alpha, \mu, \phi) = \frac{k}{\phi} \exp \left( -k \frac{x - \mu}{\phi} \right) \left( 1 + \frac{1}{\alpha} \right) ^{\frac{1}{\alpha} (\alpha + 1)} , \quad \alpha > 0 , \quad \text{and} \quad \phi > 0 . \quad \alpha \text{ is skewness parameter.}
\]

\( k \) is spread parameter, \( \mu \) is location parameter, and \( \phi \) is a function of \( \alpha \). Bayesian methods, therefore, proposed to be a unique procedure for solving the parameters estimation. To do so, the next step is how to estimate the posterior of each parameter density numerically.

Estimating parameter in neo-normal, which has CDF and pdf in (1) and (2), are firstly done by implementing it to WinBUGS by following Lunn and Jackson (2004). The code of this module is listed in Appendix A. Illustration of this algorithm for estimating neo-normal density is shown as Directed Acyclic Graph (DAG) as in Figure 1. Node “y[i]” is the data whose density to be estimated, node “alpha” is skewness parameter and node “phi” is spread parameter are both have to be set to have strictly positive distribution. The complete form for estimating neo-normal density using Bayesian can be shown as
The posterior of neo-normal density is
\[ p(\Theta | X) \propto L(X | \Theta) \times p(\Theta) \tag{3} \]
where \( \Theta = (\mu, \phi, \alpha) \), then the likelihood \( L(X | \Theta) \) is
\[ L(x | \mu, \phi, \alpha) = \prod_{i=1}^{k} \exp \left( -k \left( \frac{x_i - \mu}{\phi} \right) \right) \left( 1 + \frac{1}{\alpha} \exp \left( -k \left( \frac{x_i - \mu}{\phi} \right) \right) \right)^{-1} \tag{4} \]

Joint posterior of \( \mu, \phi, \) and \( \alpha \) with independent prior \( p(\alpha), p(\mu), \) and \( p(\phi) \) is
\[ p(\mu, \phi, \alpha | x) \propto p(\alpha) p(\mu) p(\phi) L(x | \mu, \phi, \alpha) \]
\[ \propto b^\alpha \lambda^{b-1} A B \frac{1}{\sqrt{2\pi} \sigma^2} \prod_{i=1}^{k} \phi \frac{C}{\left( 1 + \frac{C}{\alpha} \right)^{a+1}} \tag{5} \]
where \( L(x | \mu, \phi, \alpha) \) is joint likelihood of \( \mu, \phi, \) and \( \alpha \):
\[ A = \exp \left( -\left( \frac{\alpha - \lambda}{a} \right)^b \right), \quad B = \exp \left( -\frac{\mu - \eta}{2\sigma^2} \right), \quad C = \exp \left( -k \left( \frac{x_i - \mu}{\phi} \right) \right), \quad \phi > 0, \quad \alpha > 0, \quad \alpha > 0. \]
\( a, b > 0, \lambda > 0, \quad p < q, \quad -\infty < x < \infty, \quad -\infty < \mu < \infty, \quad -\infty < \eta < \infty, \quad p \leq x \leq q \) and
\[ k = \frac{1}{\sqrt{2\pi}} \left( 1 + \frac{1}{\alpha} \right)^{a+1} \]
Equation (5) will be treated as full conditional form of each individual parameter then implemented in WinBUGS. The step is as follow:
Algorithm 1: Steps for estimating neo-normal density

1. Determine the initial value of \( \mu^0, \phi^0, \) and \( \alpha^0 \) also set \( i = 1 \)

2. Generate \( \mu^{(i)} \) from \( \mu|\phi^{(i-1)}, \alpha^{(i-1)}, x \sim B(n, k) \frac{k}{\phi^{(i-1)}} \frac{C}{\left(1 + \frac{C}{\alpha^{(i-1)}}\right)^{\phi^{(i-1)}+1}} \) \hspace{1cm} (6)

3. Generate \( \phi^{(i)} \) from \( \phi|\mu^{(i-1)}, \alpha^{(i-1)}, x \sim \prod_{i=1}^{n} \frac{k}{\phi^{(i-1)}} \frac{C}{\left(1 + \frac{C}{\alpha^{(i-1)}}\right)^{\phi^{(i-1)}+1}} \) \hspace{1cm} (7)

4. Generate \( \alpha^{(i)} \) from \( \alpha|\mu^{(i-1)}, \phi^{(i-1)}, x \sim \alpha - \lambda \prod_{i=1}^{n} \frac{k}{\phi^{(i-1)}} \frac{C}{\left(1 + \frac{C}{\alpha^{(i-1)}}\right)^{\phi^{(i-1)}+1}} \) \hspace{1cm} (8)

When this neo-normal is used for representing the error of linear model, then the estimating parameter is just by changing parameter \( \mu \) or node “mu[i]” with \( E[Y] \) or linear combination of \( X, \) or \( \beta X. \) The change of Figure 1 can be seen in Figure 5 in section 4.

4. Numerical Implementation

Two implementations are demonstrated here for showing the works of density estimation and linear model estimation. For density estimation, the small sample generated data in Table 1 would be estimated to be neo-normally distributed. The descriptive shows that data is skewed to the left. By using DAG in Figure 1, after WinBUGS is iterated 10000 times, the estimated skewness parameter of \( \alpha \) is 0.6264. It is that the model has \( \alpha < 1 \) and gives information that the data is detected to be skewed to the left as shown in Figure 2. WinBUGS output is shown in Table 2 and the posterior density is exposed in Figure 3.

The second implementation is applying neo-normal error in modelling rice production in East Java during the year 2000. The rice production is supposed to be influenced by area of harvest, area of irrigated area of failed harvest, and the number of raining days. The data has been analyzed using Ordinary Square and only one independent variable is significant with \( R^2 = 98.6\%, \) that is area of harvest. The plot of error (Figure 4), however, is still show non-normally distributed and tends to be skewed to the left. After the data is modelled using neo-normal linear model via WinBUGS, the result shows that \( \alpha \) is 0.244. The model is accurately estimated by WinBUGS and is relatively better than OLS.

Table 1: Simulated data of Neo-Normal

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>3</th>
<th>5</th>
<th>5</th>
<th>6</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>11</td>
<td>11</td>
<td>12</td>
</tr>
</tbody>
</table>

Figure 2: Histogram of data in Table 1.
Tabel 2: WinBUGS output for estimating neo-normal density

<table>
<thead>
<tr>
<th>node</th>
<th>alpha</th>
<th>mu</th>
<th>phi</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>0.6264</td>
<td>6.265</td>
<td>8.566</td>
</tr>
<tr>
<td>sd</td>
<td>0.1636</td>
<td>0.4227</td>
<td>1.28</td>
</tr>
<tr>
<td>MC_error</td>
<td>0.002024</td>
<td>0.01278</td>
<td>0.01924</td>
</tr>
<tr>
<td>2.5%</td>
<td>0.2842</td>
<td>5.425</td>
<td>6.527</td>
</tr>
<tr>
<td>median</td>
<td>0.6377</td>
<td>6.28</td>
<td>8.403</td>
</tr>
<tr>
<td>97.5%</td>
<td>0.9088</td>
<td>7.084</td>
<td>11.48</td>
</tr>
<tr>
<td>start</td>
<td>4001</td>
<td>4001</td>
<td>4001</td>
</tr>
<tr>
<td>sample</td>
<td>6000</td>
<td>6000</td>
<td>6000</td>
</tr>
</tbody>
</table>

Figure 3: Estimated posterior density of $\mu$, $\phi$, and $\alpha$.

Figure 4: Plot of error using OLS

DAG for this model is given in Figure 5. After WinBUGS code of this DAG is run for 100,000 times, the result is shown in Figure 6. All independent variables are significant with the linear model as in (9) with $\text{MSE} = 1109900$, which is much smaller than $\text{MSE}$ form OLS ($\text{MSE} = 293262170$).
Rice Production(ton) = \[-16330 + 1.241 \times \text{Area_of_harvest (Ha)}
+ 0.1927 \times \text{Area_ irrigation (Ha)}
- 21.77 \times \text{Area_failed_harvest}
+ 107.1 \times \text{number_of_raining_days} \quad (9)\]

**Figure 5:** DAC for rice production in East Java

<table>
<thead>
<tr>
<th>node</th>
<th>mean</th>
<th>sd</th>
<th>5% error</th>
<th>25%</th>
<th>median</th>
<th>75%</th>
<th>start</th>
<th>sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>k[i]</td>
<td>1.241</td>
<td>0.0519</td>
<td>0.02201</td>
<td>-0.05757</td>
<td>1.244</td>
<td>2.322</td>
<td>4001</td>
<td>98000</td>
</tr>
<tr>
<td>T[J]</td>
<td>0.1927</td>
<td>1.001</td>
<td>0.016373</td>
<td>-1.831</td>
<td>0.2001</td>
<td>2.15</td>
<td>4001</td>
<td>96000</td>
</tr>
<tr>
<td>T[G]</td>
<td>-21.77</td>
<td>0.2258</td>
<td>0.001694</td>
<td>-22.22</td>
<td>-21.77</td>
<td>-23.33</td>
<td>4001</td>
<td>95000</td>
</tr>
<tr>
<td>T[H]</td>
<td>107.1</td>
<td>0.9429</td>
<td>5.46894</td>
<td>106.9</td>
<td>107.1</td>
<td>107.3</td>
<td>4001</td>
<td>96000</td>
</tr>
<tr>
<td>const</td>
<td>-16330.0</td>
<td>0.01584</td>
<td>5.7526</td>
<td>-16340.0</td>
<td>-16340.0</td>
<td>-16330.0</td>
<td>4001</td>
<td>96000</td>
</tr>
<tr>
<td>done</td>
<td>1.875</td>
<td>0.4047</td>
<td>0.001731</td>
<td>0.8756</td>
<td>1.906</td>
<td>2.734</td>
<td>4001</td>
<td>96000</td>
</tr>
<tr>
<td>tau</td>
<td>111330.0</td>
<td>948.6</td>
<td>4.882</td>
<td>108400.0</td>
<td>111200.0</td>
<td>1,12545</td>
<td>4001</td>
<td>96000</td>
</tr>
</tbody>
</table>

**Figure 6:** WinBUGS output after 100000 iterations

**Conclusion**

Neo-normal distribution that has succeed to be implemented as an add-ins in WinBUGS 1.4 is possible to be used for identifying neo-normal density of data and is relatively improve the OLS in estimating model.

**Acknowledgements**

This research was sponsored by DIPA Institut Teknologi Sepuluh Nopember (ITS), Surabaya, operated by the Research and Community Services Institute of ITS for the year of 2009, contract number: 74/12.7/PM/2009. The author wishes to thank the anonymous referees for their valuable comments which led to considerable improvement in this article and also give chance for presenting this paper in ICoMS-4 conference.
Neu-Normal Module in WinBUGS

MODULE WBDevNeoNormalX;

IMPORT
WBDevUnivariate,
WBDevRandom,
WBDevSpecial,
Math;

CONST
location := 0;
dispersion := 1;
skewness := 1;

TYPE
StdNode = POINTER TO RECORD (WBDevUnivariate, StdNode) END;
Left = POINTER TO RECORD (WBDevUnivariate, Left) END;
Right = POINTER TO RECORD (WBDevUnivariate, Right) END;
Interval = POINTER TO RECORD (WBDevUnivariate, Interval) END;
Factory = POINTER TO RECORD (WBDevUnivariate, Factory) END;

VAR

log2Pi: REAL;

fact: WBDevUnivariate.Factory;

PROCEDURE DeclareArgTypes (OUT args: ARRAY OF CHAR);
BEGIN
args := "xxx";
END DeclareArgTypes;

PROCEDURE DeclareProperties (OUT isDiscrete, canIntegrate: BOOLEAN);
BEGIN
isDiscrete := FALSE;
canIntegrate := TRUE;
END DeclareProperties;

PROCEDURE NaturalBounds (node: WBDevUnivariate.Node; OUT lower, upper: REAL);
BEGIN
lower := -INF;
upper := INF;
END NaturalBounds;

PROCEDURE LogFullLikelihood (node: WBDevUnivariate.Node; OUT value: REAL);
VAR
x, skew, dis, mu, k, temp: REAL;
BEGIN
x := node.value;
mu := node.arguments[location][0].Value();
dis := node.arguments[dispersion][0].Value();
skew := node.arguments[skewness][0].Value();

k := (1/Math.Sqrt(2*Math.Pi()))*(Math.Power(1+(1/skew), skew+1));
temp := Math.Exp(-k*((skew-mu)/dis));
value := Math.Log(k/dis)*k*((x-mu)/dis));
value := value-{skew+1}*Math.Log((skew+temp)/skew);

END LogFullLikelihood;

PROCEDURE LogPropLikelihood (node: WBDevUnivariate.Node; OUT value: REAL);
BEGIN
LogFullLikelihood(node, value);
END LogPropLikelihood;
PROCEDURE LogPrior (node: WBDcvUnivariate.Node; OUT value: REAL);
VAR
x,skew,dis,mu,k: REAL;
BEGIN
x := node.value;
mu := node.arguments[location][0].Value();
dis := node.arguments[dispersion][0].Value();
skew := node.arguments[skewness][0].Value();
k := (1/Math.Sqrt(2*Math.Pi()))*(Math.Power(1+(1/skew), skew+1));
value := Math.Exp(-k*((skew-mu)/dis));
END LogPrior;

PROCEDURE Cumulative (node: WBDcvUnivariate.Node; x: REAL; OUT value: REAL);
VAR
skew,dis,mu,k,temp: REAL;
BEGIN
mu := node.arguments[location][0].Value();
dis := node.arguments[dispersion][0].Value();
skew := node.arguments[skewness][0].Value();
k := (1/Math.Sqrt(2*Math.Pi()))*(Math.Power(1+(1/skew), skew+1));
temp := Math.Exp(-k*((skew-mu)/dis));
value := Math.Power(1+temp/skew,-skew);
END Cumulative;

PROCEDURE DrawSample (node: WBDcvUnivariate.Node; censoring: INTEGER; OUT sample: REAL);
VAR
skew,dis,mu, left, tau, right: REAL;
BEGIN
mu := node.arguments[location][0].Value();
dis := node.arguments[dispersion][0].Value();
skew := node.arguments[skewness][0].Value();
node.Bounds(left, right);
CASE censoring OF
| WBDcvUnivariate.noCensoring:
sample := WBDcvRandnum.NeoNorm(mu,dis,skew);
| WBDcvUnivariate.leftCensored:
sample := WBDcvRandnum.NeoNorm(mu,dis,skew);
| WBDcvUnivariate.rightCensored:
sample := WBDcvRandnum.NeoNorm(mu,dis,skew);
| WBDcvUnivariate.intervationCensored:
sample := WBDcvRandnum.NeoNorm(mu,dis,skew);
END;
END DrawSample;

VAR
node: WBDcvUnivariate.Node;
stdNode: StdNode; left: Left; right: Right; interval: Interval;
BEGIN
CASE option OF
| WBDcvUnivariate.noCensoring:
    NW(stdNode);
    node := stdNode;
| WBDcvUnivariate.leftCensored:
    NW(left);
    node := left;
| WBDcvUnivariate.rightCensored:
    NW(right);
    node := right;
END;

| WBDevUnivariate.intervalCensored; |
| NEW(interval); |
| node := interval; |
| END; |
| node.SetCumulative(Cumulative); |
| node.SetDeclareArgTypes(DeclareArgTypes); |
| node.SetDeclareProperties(DeclareProperties); |
| node.SetDrawSample(DrawSample); |
| node.Set.logFull.likelihood(LogFull.likelihood); |
| node.Set.logProp.likelihood(LogProp.likelihood); |
| node.SetLogPrior(LogPrior); |
| node.SetNaturalBounds(NaturalBounds); |
| node.Initialize; |
| RETURN node; |

PROCEDURE Instal**;
BEGIN |
WBDevUnivariate.Install(fit); |
END install;

PROCEDURE Init;
VAR |
f: Factory; |
BEGIN |
log2Pi := Math.Ln(2 * Math.Pi()); |
NEW(f); fact := f; |
END Init;

BEGIN |
Init; |
END WBDevNeoNormalX.

References

[13] Lunn, D. and Jackson, C., 2004. WinBUGS Development Interface (WBDev) – Implementing your own univariate distributions, Imperial College School of Medicine, London, UK


ON ADDING UNIVARIATE DISTRIBUTIONS IN WINBUGS USING BLACK-BOX COMPONENT BUILDER

Dedy Dwi Prastyo, Nur Iriawan, and Dwi Yulianto
Statistics Department, Faculty of Mathematics and Natural Sciences, Institut Teknologi Sepuluh Nopember, Surabaya - Indonesia

Abstract. Bayesian approaches has broadly used in many field of applications, especially when the data limited and the analytical approaches lead to be more difficult. WinBUGS 1.4 as open software for Bayesian tools are frequently used for data analysis by Bayesians people. Supposing some models which could be analyzed using Bayesian approaches sometimes WinBUGS cannot help. It is due to some distributions related to its full conditional posterior of that case has not been compiled in WinBUGS 1.4. This paper demonstrates the use of Black-Box component builder to add WinBUGS library with some distributions, therefore WinBUGS could be ready for analyzing posterior density, bayesian regression modeling, and other bayesian approaches for data analysis.

Keywords: Bayesian, Univariate Distribution; Bayesian Regression Modeling.

1. Introduction

Many researchers develop statistical distributions and statistical methods; especially incorporate with goodness of fit of data and diagnostic checking of residual in modelling. Those researches involve the conventional approaches in analysis and modelling data, that is the residual of the observation is assumed as IID (0, $\sigma^2$) (Independent and Identically Distributed Normal) [Iriawan, 1999].

Recently, many researchers emphasize the development of statistical distribution using bayesian approach. Bayesian approach expand the normality assumptions of residual in the statistical modeling. In many cases, residual of the statistical model can be skew or not normal. The distribution could be exponential power [Box and Tiao, 1973], Neo-Normal [Iriawan, 1999], Skew-normal and Skew-student [Fernandes and Steel, 1998]. Neo-normal, skew-normal, and skew-student have skewness parameter that control and identify the skew property of data. Exponential power could model the data which has the property leptocurtic or symmetric with fat tail.

WinBUGS is the current, windows-based, version of the BUGS (Bayesian Inference using Gibbs Sampler) software. It is a user-friendly, ‘point-and-click’ environment that makes accessible state-of-the-art statistical methodology (Markov Chain Monte Carlo technique) for analysis of a wide class of bayesian full probability models [Lunn, et al., 2000].

The conceptual design of the software is based on constructing an internal representation of the probability model that is analogous to the way in which it may be visualized as a graphical model. In graphical modeling, each quantity in the model is represented by a node and nodes are connected by lines or arrows to show direct dependence. The details of distributional assumptions and deterministic relationships are ‘hidden’ to clarify the qualitative nature of the model. This has led very naturally to an object-oriented approach to the software’s design.

WinBUGS has been designed primarily for handling directed acyclic graph (DAG), graphical models were links between nodes are directed and cycles are not permitted. DAGs represents a series of conditional independence assumptions, which allow the full probability model to be factorized into product of simple local components.

The structure of the WinBUGS source-code is also analogous to a graphical model, in that it comprises a network of locally communicating components—a component-oriented philosophy has been adopted. This novel software engineering approach aims to create fully extensible modular system.
Software consist of number of component that are not linked together until load-time or even run-time. Each software component has well-defined interface that describes the implemented entities that can be used in other components. The component interface is encoded in machine readable format called a ".bol file", the use of which allows consistency of the component interfaces to be checked both at compile-time and link-time, thus improving the reliability of the software. Including of new methods and functions is achieved by writing extra component that simply either "plug-in" to relevant slots in testing modules, or make use of existing modules, without requiring any part of the software to be compiled [Lunn, et al., 2000].

Some distributions such as: neo-normal, skew-normal, and skew-student are not available in the WinBUGS. It is needed to add those distributions to WinBUGS in order to expand the use of WinBUGS solve many cases without any restricted assumption. This aim of this paper is to describe how to add univariate distribution (skew-student and exponential power) to WinBUGS using Black-Box Developer Builder. The added distribution will be used in regression modeling using bayesian approach.

2 Skew-Student Distribution

Skew-Student is Student-t distribution with skew parameter which is developed by Fernandez dan [1998]. A random variable $\varepsilon$ (with skew parameter $\gamma$ is known) has probability [Fernandez dan 1998].

$$p(\varepsilon | \gamma) = \frac{2}{\gamma^+} \left\{ \frac{f(\varepsilon)}{\gamma} I_{(0,\gamma)}(\varepsilon) + f(\gamma \varepsilon) I_{(-\infty,0)}(\varepsilon) \right\}$$

$$-\infty < \varepsilon < 0, \gamma > 0$$

Student-t distribution in formulae (3) is substituted to formulae (2) for getting Skew-Student distribution formulae (4). Representation of Skew-Student distribution with degree of freedom $\nu=10$, skewness parameter $\gamma < 1$ and $\gamma > 1$ can be seen in Figure 1a and Figure 1b.

$$f(x | \nu) = \frac{\Gamma \left( \frac{\nu+1}{2} \right)}{\Gamma (\nu/2) \sqrt{\nu \pi}} \left( 1 + \frac{x^2}{\nu} \right)^{-\frac{\nu+1}{2}}$$

$$f(x | \nu, \gamma) = \frac{2}{\gamma^+} \frac{\Gamma \left( \frac{\nu+1}{2} \right)}{\gamma^+} \left( 1 + \frac{x^2}{\nu} \right)^{-\frac{\nu+1}{2}}$$

$$-\infty < x < \infty, \nu = 1, 2, 3, \ldots \text{ and } \gamma > 0$$

![Figure 1. pdf of Skewed-Student distribution with $\nu = 10$, (a) with $\gamma < 1$; (b) with $\gamma > 1$](image)
3. Maximum Likelihood Estimation

Model \( y_i = \beta_0 + \beta_1 x_i + \epsilon_i \) and \( \epsilon_i \) is assumed normally distributed with mean \( \mu = 0 \) and variance \( \sigma^2 = \text{Var}(\epsilon_i) \). Estimated mean and variance are derived from likelihood of \( \epsilon_i \):

\[
L = \prod_{i=1}^{n} f(\epsilon_i) = \frac{1}{(2\pi)^{n/2}\sigma^n} \exp \left( -\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2 \right)
\]

(5)

Estimated parameter \( \beta_0 \) and \( \beta_1 \) in (6) are partially derived from log likelihood function in (3) [Draper and Smith, 1992]:

\[
\hat{\beta}_1 = \frac{\sum_{i=1}^{n} x_i y_i - n \overline{xy}}{\sum_{i=1}^{n} x_i^2 - n \overline{x^2}}
\]

(6.a)

\[
\hat{\beta}_0 = \overline{y} - \hat{\beta}_1 \overline{x}
\]

(6.b)

Likelihood function for model \( y_i = \beta_0 + \beta_1 x_i + \epsilon_i \) with \( \epsilon_i \) is assumed skewed-student distributed is shown in (7). Estimation parameters from MLE in (7) is very difficult. For handling this difficulty, Bayesian approach is used for estimating the parameters of regression model.

\[
L = \prod_{i=1}^{n} f(\epsilon_i) = \prod_{i=1}^{n} \left( \frac{\Gamma \left( \frac{\nu+1}{2} \right)}{\Gamma \left( \frac{\nu}{2} \right) \sqrt{\pi \nu}} \right)^{\frac{1}{2}} \left[ 1 + \frac{\epsilon_i^2}{\nu} \right]^{-\frac{\nu+1}{2}}
\]

(7)

4. Bayesian Statistics

Suppose we have observed data \( y \) and unknown parameter \( \theta \). The Bayesian approach to statistics is to treat all unknown quantities as random variables and assign a prior probability distribution to each. By also specifying a joint probability distribution for the data, i.e. a likelihood, we obtain a full probability model for all observable and unobservable quantities. In order to make inferences about \( \theta \), we use Bayes' theorem to construct the posterior distribution, i.e. the joint distribution of all model parameters conditional on the observed data:

\[
p(\theta | y) \propto l(y | \theta)p(\theta)
\]

(8)

where \( l(y | \theta) \) is likelihood of the data and \( p(\theta) \) is prior distribution. An excellent introduction to Bayesian data analysis is given by Gelman et al. [1995].
5. Creating Module in WinBUGS

The Black-Box Component Builder provide a user interface that enables rapid application development: it incorporates a powerful text editor, for example, and to some extent automates the programming of many of the essential features of modern software. WinBUGS has been designed so that users can extend it, with minimal effort, to meet their own requirements. All that is required is some familiarity with the Component Pascal language and structure of the WinBUGS framework. There are three main ways of extending the framework: (i) new types of logical node; (ii) new types of stochastic node; (iii) new MCMC updating algorithms [Lunn, et al., 2000].

Steps of adding new module in WinBUGS is as follow:

- Download BlackBox Component Builder from http://www.oberon.co/blackbox.html
- unig BlackBox file by clicking icon SetupBlackBox.exe (put in C:\Program Files\BlackBox Component Builder 1.5)
- Copy (Ctrl+C) all files in directory of WinBUGS14 (Program Files\WinBUGS14) and Paste (Ctrl+V) them to BlackBox directory (Program Files\BlackBox). Choose “Yes to All” to replace all files already in BlackBox directory.
- Now WinBUGS 14 and BlackBox are ready to be used naturally for Bayesian Modeling and developing modules.
- Download additional component wbdev_shared_source.exe and wbdev.exe from http://www.winbugs-development.org.uk and the extract those files to Program Files\BlackBox Component Builder 1.5.
- Open file: Program Files\BlackBox Component Builder 1.5\wbdev_shared_code.txt using BlackBox, then decode the file. Do the same way to file wbdev_01_09_04.txt. Close Black-Box and open again to develop WinBUGS.

The modul can be developed by modifying Univariate Template module (see in Program Files\BlackBox\WBDev\Mod\UnivariateTemplate.odc).

6. Application

Skew Student distribution will be applied for modelling the relationship of Anomaly of Square Harvest (AnLP) with weighted rainfall (WRI) in District Subang in period III (September-December) 1992 - 2006. In 1992-2006, the average of AnLP of paddy in period III is 1.488 Ha. In 2003-2005 AnLP increase, but in 2006 AnLP decrease (Fig. 2). El-Nino in March 1997-April 1998 caused the AnLP increase.

```
\text{AnLP} = 1.1863 + 1.3972 \text{WRI}_9 - 0.3604 \text{WRI}_10 + 0.0491 \text{WRI}_11 + 0.0028 \text{WRI}_12
```

Figure 2. Trend of AnLP in District Subang Periode III (September-December).

Figure 3. Distribution of Residual.
Normality test of residual using Kolmogorov-Smirnov result: p-value = 0.053 (near to α=0.05). Histogram of residual (Fig. 3) shows that the distribution of residual is not symmetric but left skewed. We make hypothesis that the residual skew-student distributed. The estimation of parameter in skew-student distribution of residual uses additional module of WinBUGS. Figure 4.a shows the doodle in WinBUGS with x[i] is residual data, degree of freedom (d) and skew parameter (s). MCMC iteration of skew parameter (s) result: mean 0.7203, median 0.7109, standard deviation 0.1289, MC error 0.001312, 95% confidence interval 0.4945 up to 0.9958.

Bayesian approach will be used for modelling the regression in (9) with residual skew-student distributed. MCMC is run 10,000 times with burnt in 4000. Regression model of AnLP and WRI with residual skewed-student distributed is (Table 1):

\[ \text{AnLP}_2 = 1.183 + 1.393 \times \text{WRI}_9 - 0.3623 \times \text{WRI}_{16} + 0.0384 \times \text{WRI}_{11} - 0.01022 \times \text{WRI}_{12} \]

(10)
Table 1. Estimation of Regression Parameter with Skewed-Student Residual

<table>
<thead>
<tr>
<th>node</th>
<th>mean</th>
<th>sd</th>
<th>MC error</th>
<th>2.50%</th>
<th>median</th>
<th>97.50%</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1.183</td>
<td>0.3315</td>
<td>0.009654</td>
<td>0.5671</td>
<td>1.182</td>
<td>1.877</td>
</tr>
<tr>
<td>b[1]</td>
<td>1.393</td>
<td>0.6749</td>
<td>0.01539</td>
<td>0.05822</td>
<td>1.384</td>
<td>2.693</td>
</tr>
<tr>
<td>b[2]</td>
<td>-0.3623</td>
<td>1.451</td>
<td>0.04249</td>
<td>-3.306</td>
<td>-0.3058</td>
<td>2.368</td>
</tr>
<tr>
<td>b[3]</td>
<td>0.0384</td>
<td>3.633</td>
<td>0.1067</td>
<td>-7.379</td>
<td>0.1752</td>
<td>6.974</td>
</tr>
<tr>
<td>b[4]</td>
<td>-0.01022</td>
<td>3.385</td>
<td>0.09712</td>
<td>-6.626</td>
<td>0.04444</td>
<td>6.782</td>
</tr>
<tr>
<td>sigma</td>
<td>0.2319</td>
<td>0.2106</td>
<td>0.006264</td>
<td>0.02826</td>
<td>0.1545</td>
<td>0.7814</td>
</tr>
<tr>
<td>skew</td>
<td>0.4825</td>
<td>0.2295</td>
<td>0.004308</td>
<td>0.1742</td>
<td>0.4416</td>
<td>1.03</td>
</tr>
</tbody>
</table>

Conclusion

From the information above, we can conclude the module in WinBUGS can be added using Black-Box Component Developer which expand the application of bayesian in many cases. The additional module in WinBUGS using Black-Box developer give valid estimation. The skew-student distribution in WinBUGS can handle statistical data analysis without any restriction in normality distribution assumption, i.e. regression modeling. The future works of this paper is the development of multivariate distribution in WinBUGS.
Appendix: Skew-Student Module in WinBUGS

MODULE WDBdevSkewT;
IMPORT
WDBdevUnivariate,
WDBdevRandnum, WDBdevSpecfunc,
Math, MathFunc;
CONST
loca=0, sebaran=1, skewness=1, df=2;
TYPE
StdNode = POINTER TO RECORD (WDBdevUnivariate.StdNode)
lambda: WDBdevUnivariate.Factor;
END;
Lef = POINTER TO RECORD (WDBdevUnivariate.Left) END;
Right = POINTER TO RECORD (WDBdevUnivariate.Right) END;
Interval = POINTER TO RECORD (WDBdevUnivariate.Interval) END;
Factory = POINTER TO RECORD (WDBdevUnivariate.Factory) END;

VAR
log2Pi: REAL;
fac: WDBdevUnivariate.Factory;
PROCEDURE DeclareArgTypes (OUT args: ARRAY OF CHAR);
BEGIN
args := "issc";
END DeclareArgTypes;
PROCEDURE DeclareProperties (OUT IsDiscrete, canIntegrate: BOOLEAN);
BEGIN
IsDiscrete := FALSE;
canIntegrate := FALSE;
END DeclareProperties;
PROCEDURE NaturalBounds (node: WDBdevUnivariate.Node; OUT lower, upper: REAL);
BEGIN
lower := -INF;
upper := INF;
END NaturalBounds;
PROCEDURE LogFullLikelihood (node: WDBdevUnivariate.Node; OUT value: REAL);
VAR
x, y, skew, mu, tau: REAL;
BEGIN
x := node.value;
y := node.arguments[0].Value();
skew := node.arguments[skewness].Value();
mu := node.arguments[Skewmu].Value();
tau := node.arguments[sebaran].Value();
value := Math.Ln(2 + skew^2 - Math.Ln(Math.Power(ske, 2) + 1)) +
WDBdevSpecfunc.LogGammaFunc((v+1)/2) -
WDBdevSpecfunc.LogGammaFunc(v/2);
value := value - 0.5 * Math.Ln(v * Math.PI());
IF x=0 THEN
value := value - ((v+1)/2) * Math.Ln((v+1) * (tau*(x-mu) / skew)^2 / tau*(x-mu)/skew) / v;
ELSE
value := value - ((v+1)/2) * Math.Ln((v+1) * (tau*(x-mu) / skew)^2 / tau*(x-mu)/skew) / v;
END;
END LogFullLikelihood;
PROCEDURE LogPrior (node: WDBdevUnivariate.Node; OUT value: REAL);
VAR
skew, v, x, mu, tau: REAL;
BEGIN
END LogPrior;
Prastyo, Irwan, and Yulianto 4th International Conference on Mathematics and Statistics 2009

\[ x := \text{node.value}; \]
\[ y := \text{node.arguments[0][0].Value();} \]
\[ \text{skew} := \text{node.arguments[skewness][0].Value();} \]
\[ \text{mu} := \text{node.arguments[loclip][0].Value();} \]
\[ \tau := \text{node.arguments[sebclip][0].Value();} \]

IF \( x = 0 \) THEN
\[ \text{value} := -0.50 \ast \text{tau} \ast (x - \mu) / \text{skew} \ast (x - \mu); \]
ELSE
\[ \text{value} := -0.50 \ast \text{tau} \ast (x - \mu) \ast \text{skew} \ast (x - \mu); \]
END;

PROCEDURE Cumulative (node: WBDevUnivariate.Node; x: REAL; OUT value: REAL);
VAR
\[ \text{skew}, y, \text{mu}, \tau, \text{value} : \text{REAL}; \]
BEGIN
\[ x := \text{node.value}; \]
\[ y := \text{node.arguments[0][0].Value();} \]
\[ \text{skew} := \text{node.arguments[skewness][0].Value();} \]
\[ \text{mu} := \text{node.arguments[loclip][0].Value();} \]
\[ \tau := \text{node.arguments[sebclip][0].Value();} \]
\[ \text{value} := (2 \ast \text{skew}) / (\text{skew}^2); \]
IF \( x > 0 \) THEN
\[ \text{value} := \text{value} + 1 - 0.5 \ast \text{WBDevSpecfunc.Beta}(0.5 \ast y, 0.5, y / (y + \text{tau} \ast x / \text{skew} \ast x \ast \text{skew})) \]
ELSE
\[ \text{value} := \text{value} + 0.5 \ast \text{WBDevSpecfunc.Beta}(0.5 \ast y, 0.5, y / (y + \text{tau} \ast x / \text{skew} \ast x \ast \text{skew})) \]
END;
END Cumulative;

PROCEDURE DrawSample (node: WBDevUnivariate.Node; censoring: INTEGER; OUT sample: REAL);
VAR
\[ x, \text{skew}, \mu, \tau, \text{left}, \text{right} : \text{REAL}; \]
BEGIN
\[ x := \text{node.value}; \]
\[ y := \text{node.arguments[0][0].Value();} \]
\[ \text{skew} := \text{node.arguments[skewness][0].Value();} \]
\[ \text{mu} := \text{node.arguments[loclip][0].Value();} \]
\[ \tau := \text{node.arguments[sebclip][0].Value();} \]
\[ \text{node.} . \text{Bounds} (\text{left}, \text{right}); \]
CASE censoring OF
\[ \text{WBDevUnivariate.noCensoring;} \]
\[ \text{sample} := \text{WBDevRandnum.Normal}(0,1); \]
\[ \text{WBDevUnivariate.leftCensored;} \]
\[ \text{sample} := \text{WBDevRandnum.Normal.LeftTail}(0,1, \text{left}); \]
\[ \text{WBDevUnivariate.rightCensored;} \]
\[ \text{sample} := \text{WBDevRandnum.Normal.RightTail}(0,1, \text{right}); \]
\[ \text{WBDevUnivariate.intervalCensored;} \]
\[ \text{sample} := \text{WBDevRandnum.Normal}(0,1); \]
END;
END DrawSample;

PROCEDURE (f. Factory) New (option: INTEGER; WBDevUnivariate.Node);
VAR
\[ \text{node: WBDevUnivariate.Node;} \]
\[ \text{stdNode: StdNode; left: Left; right: Right; interval: Interval;} \]
BEGIN
CASE option OF
\[ \text{WBDevUnivariate.noCensoring;} \]
\[ \text{New(stNode}; \]
\[ \text{node} := \text{stNode}; \]
\[ \text{WBDevUnivariate.leftCensored;} \]
\[ \text{New(left}; \]
\[ \text{node} := \text{left}; \]
\[ \text{WBDevUnivariate.rightCensored;} \]
\[ \text{New(right}; \]
\[ \text{node} := \text{right}; \]
\[ \text{WBDevUnivariate.intervalCensored;} \]
\[ \text{New(interval}; \]
\[ \text{node} := \text{interval}; \]
PROCEDURE Installs;
BEGIN
WBDevUnivariate.Install(fact);
END Installs;

PROCEDURE Init;
VAR
f: Factory;
BEGIN
log2Pi := Math.Log(2 * Math.PI());
NEW(f); fact := f;
END Init;
BEGIN
Init;
END WBDevSkewT.

References


ON ESTIMATING DENSITY OF KOTA MATARAM HOUSEHOLD EXPENDITURE BASED ON COST OF LIVING SURVEY (SBH) 2007 USING BAYESIAN MIXTURE OF MIXTURE

Yassinta Ben Katarti L D1, Nur Irwan2, dan Satwiko Darmesto3
1 Mahasiswa Magister Statistika, Institut Teknologi Sepuluh Nopember Surabaya, Indonesia
2 Jurusan Statistika, Institut Teknologi Sepuluh Nopember Surabaya, Indonesia
3 Badan Pusat Statistik, Jakarta, Indonesia

Abstract. The latest Living Cost Survey (SBH) in Indonesia was held in 2007 in 66 different municipalities, one of them is Kota Mataram. SBH is used to construct the weighted diagram for composing the measurement of Consumers Price Index (IHK) and for measuring National Income. IHK, furthermore, is used for calculating inflation. Another output of SBH is the municipality household expenditure, where its patterns tend to be multimodal and has fat-tails in the right side. This research considers to finding step of constructing SBH pattern which is incorporated by several commodities and their sub-commodities. Mixture model couple with Bayesian MCMC is proposed to be used here to estimate SBH pattern. By using this pattern, the density of household expenditure can be explained. Because of the complexity of SBH structure, the special mixture of mixture model will be employed.

Keywords: Bayesian Analysis, Mixture model, mixture of mixture, Markov Chain Monte Carlo (MCMC).

1. Introduction

The Living Cost Survey (SBH) in Indonesia which is held every five years by Statistics of Indonesia (BPS) is a way of government to see changes in society’s consumption pattern that always grows rapidly. This pattern changes could be triggered by several factors like the changes in the pattern of income per capita, changes in the pattern of supply and demand of goods and services, the changes of quantity and quality of goods and services, also because of behaviour changes of the society as a result of development especially in technology and communication. In 2007 BPS conducted SBH all together in 66 different municipalities in Indonesia, one of them is Kota Mataram. By using the distribution of raw data of SBH, the pattern changes of Kota Mataram’s household expenditure could be discovered. The model will show the contribution of each sub-commodities towards the commodities and also show the contribution of each commodities in generating the household expenditure based on the distribution of parent.

Mixture model is a well-known model in Bayesian Analysis. This model is often use to describe complex statistical problems (Bernardo and Giron, 1988). Mixture Analysis has had a remarkable burst of development in the last few years. The widespread availability of flexible Markov Chain Monte Carlo (MCMC) methods for the bayesian analysis of complex statistical models has led to their application to finite mixtures, whose likelihood estimation would face some difficulties. The bayesian approaches for these models are new, general and very powerful (Aitkin, 2001). Mixture distributions comprise a finite or infinite number of components, possibly of different distributional types, that can describe different features of data. They thus facilitate much more careful description of complex systems (Marin, Mengersen and Robert, 2005).

To see the contribution of each sub-commodities in such commodity, the mixture model will be used here. Then, this mixture would be used to model the contribution of commodities in generating the household expenditure. The density of household expenditure, therefore, would be generated by two steps mixture model, called mixture of mixture distribution. To estimate the parameter of the model we use Markov Chain Monte Carlo (MCMC) methods especially Gibbs Sampler.
2. Mixture Model

Bernardo dan Giron, (1988) describe mixture model as a probabilistic model which is described by the density

\[ p(x | \lambda, \theta) = \sum_{j=1}^{k} \lambda_j p(x | \theta_j), \]

where \( \lambda = \lambda_1, \lambda_2, \ldots, \lambda_k \) and \( \theta = \theta_1, \theta_2, \ldots, \theta_k \) and \( k \) denotes the number of mixands in the mixture. In this model, \( p(x | \theta_j) \) describes the probabilistic mechanism of generating data \( x \) within population \( P_j \), which is completely identified by its corresponding parameter \( \theta_j \), and \( \lambda_j \) denotes the proportion of mixands, where \( 0 \leq \lambda_j \leq 1 \), \( \sum_{j=1}^{k} \lambda_j = 1 \). The model described above is defined as finite mixture model by \( k \) numbers of mixands or components.

3. Mixture of Mixture Model

The general mixture model with \( l \) number of component mixture of mixture has the form:

\[ f(x | \theta, \pi) = \sum_{i=1}^{l} \pi_i f_i(x | \theta_i, \lambda_i) \]

where \( f(x | \theta, \pi) \) is the density function of mixture of mixture, \( f_i(x | \theta_i, \lambda_i) \) is the density function of mixture model from the \( i \)-th commodities with parameter \( \theta_i \). Thus, \( l \) denotes numbers of mixands' components of mixture of mixture, and \( \pi_i \) is the parameter of proportion of the \( i \)-th commodities that contribute to the mixture of mixture model, where \( 0 \leq \pi_i \leq 1 \), \( \sum_{i=1}^{l} \pi_i = 1 \).

The ability of mixture analysis as a method on data with mixture characteristics shows its advantages compared to the other statistical methods. Several data characteristics that seem to have opposite nature can be analyze and the pattern can be describes from the formed model (Iriawan, 2003).

4. Mixture of Mixture Estimation with MCMC

Posterior model that needs difficult integration process will also leads to difficulties in finding the marginal posterior of the parameters. Therefore, MCMC with its numerical approach becomes the best solution. Because there are numbers of commodities with different distributions that generates the household expenditure, we use 2 components (sub-commodities) which mix in a mixture model that generate the mixture of commodities as an example. Suppose that those 2 sub-commodities have Normal distributions each, then the mixture model would be:

\[ f(x | \lambda, \mu, \sigma^2) = \sum_{i=1}^{2} \lambda_i f_i(x | \mu_i, \sigma^2_i) \]

where \( 0 \leq \lambda_i \leq 1 \), \( \sum_{i=1}^{2} \lambda_i = 1 \).

The mixture function above will have the likelihood:

\[ L_{mix} = \prod_{j} \sum_{i} \lambda_i f(x_j | \mu_i, \sigma^2_i) \]

To find the posterior we choose priors for \( \mu \sim N(\xi, \tau) \), for \( \sigma^2 | \tau \sim G(\alpha, \beta) \), and for
\[ \lambda \propto D(\delta_1, \delta_2, \ldots, \delta_j) \]. \( N \) denotes Normal distribution, \( G \) denotes Gamma distribution, and \( D \) denotes Dirichlet distribution. Each parameters of each prior is supposed to be known. Thus, the joint posterior model can be computed by simply multiplying the likelihood with the prior above (Iriawan, 2001). Mixing of mixture in equation (3) with another mixture model will produce mixture of mixture, as explained earlier.

Of course the mixture of mixture model is very complicated. Estimating parameter for this model would be impossible to be done analytically. Bayesian methods have promised that able to solve this by individually as a sequence of estimating univariate model. This method called Markov Chain Monte Carlo (MCMC). Every MCMC step of iteration will produce a data vector \( \theta^{(n)} \) which renew \( \theta^{(i)} \) by following the stochastic nature of Markov chain process. To assure the convergence and stationary of the generated mixture’s parameters data, \( M \) first iteration will be omitted. This condition is known as the condition of burn-in. Thus, data that are used to estimate each parameter are data generated after this burn-in condition ((Brooks dan Roberts, 1997), (Athreya, Doss, dan Sethuraman, 1996), and (Cowles, dan Carlin, 1996)).

### Gibbs Sampler Algorithm For Equation (1).

**Step 1 (Initialization)**
Choose \( \theta^{(0)} = (\lambda^{(0)}, \mu^{(0)}, \text{and } \sigma^{(0)}) \).

**Step 2**
for \( j = 1 \) to \( T+N \)

(a) (Update) \( \theta^{(j)} = \theta^{(j-1)} \)

(b) Sequentially do the next steps (Generation)

- Generate \( \lambda^{(j)} \) from \( p(\lambda | \mu, \sigma, x) \propto D(\delta + n_1, \delta + n_2, \ldots, \delta + n_k) \)
- Generate \( \mu_{i}^{(j)} \) from \( p(\mu_i | \mu, \lambda, \sigma, x) \propto N(c_i; c; c_j)' \) (Iriawan, 2000),
  where \( c_1 \text{ and } c_2 \) constants and \( i = 1, 2, \ldots, k \)
- Generate \( \sigma_i^{(j)} \) from \( p(\sigma_i^{-2} | \sigma^{-2}, \lambda, \mu, x) \propto G_{\alpha, \beta} \)
  for \( i = 1, 2, \ldots, k \)

### 5. Implementation on Household Expenses of Kota Matarum

Computational implementation is done for modeling sub-commodities as a mixture of mixture. Each sub-commodities distribution is identified. Then, these distributions with their parameters are entered to WinBUGS 1.4 package, to build the program code. Each commodity has 1620 samples. There are 41 sub-commodities. The code is then run using all of the data until 10,000 times iterations. This program with these huge data is run in a computer of Intel Pentium dual-core T3200 (2.0 GHz, 667 MHz FSB, 1 MB L2 cache) processor. The time consuming to finish the process for this algorithm is 993 seconds. The result of the estimation of each mixture model density and mixture of mixture density with their parameters are shown in Table 1.

### Table 1. Density Estimation for Mixture of Mixture of Kota Matarum Household Expenditure

<table>
<thead>
<tr>
<th>Mixture of Food Stuff Commodities</th>
<th>Density Estimation of Foodstuff Commodities Mixture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mixture of Food Stuff Commodities</td>
<td>mean</td>
</tr>
<tr>
<td>P[1]</td>
<td>0.08302</td>
</tr>
<tr>
<td>P[2]</td>
<td>0.08325</td>
</tr>
<tr>
<td>P[3]</td>
<td>0.08388</td>
</tr>
<tr>
<td>P[4]</td>
<td>0.08422</td>
</tr>
<tr>
<td>P[5]</td>
<td>0.08291</td>
</tr>
<tr>
<td>P[6]</td>
<td>0.08268</td>
</tr>
<tr>
<td>P[7]</td>
<td>0.08421</td>
</tr>
<tr>
<td>P[8]</td>
<td>0.08351</td>
</tr>
<tr>
<td>P[9]</td>
<td>0.08274</td>
</tr>
<tr>
<td>P[10]</td>
<td>0.08262</td>
</tr>
<tr>
<td>P[11]</td>
<td>0.08406</td>
</tr>
<tr>
<td>P[13]</td>
<td>0.08287</td>
</tr>
</tbody>
</table>


### b: Density Estimation of Ready to serve Food, Drink, Cigarette, and Tobacco Commodities Mixture

| Mixture of Ready to serve Food, Drink, Cigarette, and Tobacco Commodities |
|-----------------|---|---|---|---|---|---|
| node            | mean | sd  | MC error 2.5% | median | 97.5% | start | sample |
| P2[1]           | 0.2481 | 0.191 | 0.001941 | 0.008344 | 0.2047 | 0.7076 | 501 | 9500 |
| P2[2]           | 0.2541 | 0.1949 | 0.00214 | 0.008731 | 0.2109 | 0.7094 | 501 | 9500 |
| P2[3]           | 0.2513 | 0.1923 | 0.001967 | 0.008264 | 0.2099 | 0.7068 | 501 | 9500 |
| P2[4]           | 0.2464 | 0.1926 | 0.001867 | 0.008228 | 0.2072 | 0.6989 | 501 | 9500 |

### c: Density Estimation of Housing, Electricity, Gas, and Power Source Commodities Mixture

| Mixture of Housing, Electricity, Water, Gas, and Power Source Commodities |
|-----------------|---|---|---|---|---|---|
| node            | mean | sd  | MC error 2.5% | median | 97.5% | start | sample |
| P3[1]           | 0.1986 | 0.1601 | 0.001751 | 0.006492 | 0.1574 | 0.5894 | 501 | 9500 |
| P3[2]           | 0.1963 | 0.1605 | 0.001715 | 0.006754 | 0.1561 | 0.5699 | 501 | 9500 |
| P3[3]           | 0.2041 | 0.1658 | 0.001841 | 0.004047 | 0.1629 | 0.6167 | 501 | 9500 |
| P3[4]           | 0.2003 | 0.163 | 0.001458 | 0.003722 | 0.161 | 0.6093 | 501 | 9500 |

### d: Density Estimation of Clothing Commodities

| Mixture of Clothing Commodities |
|-----------------|---|---|---|---|---|---|
| node            | mean | sd  | MC error 2.5% | median | 97.5% | start | sample |
| P4[1]           | 0.2526 | 0.1963 | 0.002043 | 0.008389 | 0.2062 | 0.7106 | 501 | 9500 |
| P4[2]           | 0.2463 | 0.1918 | 0.001843 | 0.007682 | 0.2046 | 0.6943 | 501 | 9500 |
| P4[3]           | 0.2504 | 0.1927 | 0.001974 | 0.008818 | 0.2099 | 0.7035 | 501 | 9500 |
| P4[4]           | 0.2507 | 0.1935 | 0.001722 | 0.009513 | 0.2084 | 0.7058 | 501 | 9500 |

### e: Density Estimation of Health Commodities

| Mixture of Health Commodities |
|-----------------|---|---|---|---|---|---|
| node            | mean | sd  | MC error 2.5% | median | 97.5% | start | sample |
| P5[1]           | 0.2478 | 0.1925 | 0.002023 | 0.00914 | 0.2058 | 0.7044 | 501 | 9500 |
| P5[2]           | 0.248 | 0.1935 | 0.001997 | 0.007592 | 0.2033 | 0.7118 | 501 | 9500 |
| P5[3]           | 0.2517 | 0.1969 | 0.001991 | 0.008209 | 0.2063 | 0.7169 | 501 | 9500 |
| P5[4]           | 0.2525 | 0.1967 | 0.001948 | 0.009279 | 0.2059 | 0.7136 | 501 | 9500 |

### f: Density Estimation of Education, Recreation, and Sport Commodities

| Mixture of Education, Recreation, and Sport Commodities |
|-----------------|---|---|---|---|---|---|
| node            | mean | sd  | MC error 2.5% | median | 97.5% | start | sample |
| P6[1]           | 0.1421 | 0.1234 | 0.001222 | 0.004121 | 0.1073 | 0.4577 | 501 | 9500 |
| P6[2]           | 0.1418 | 0.1235 | 0.001228 | 0.004050 | 0.1093 | 0.4584 | 501 | 9500 |
| P6[3]           | 0.1427 | 0.1243 | 0.001305 | 0.004415 | 0.107 | 0.4864 | 501 | 9500 |
| P6[4]           | 0.1414 | 0.1217 | 0.001223 | 0.004149 | 0.1087 | 0.4515 | 501 | 9500 |
| P6[5]           | 0.1411 | 0.1246 | 0.001244 | 0.004175 | 0.1089 | 0.4576 | 501 | 9500 |
| P6[6]           | 0.1425 | 0.1243 | 0.001183 | 0.003965 | 0.1075 | 0.4626 | 501 | 9500 |
| P6[7]           | 0.1439 | 0.1248 | 0.001328 | 0.004045 | 0.1095 | 0.4621 | 501 | 9500 |

### g: Density Estimation of Transportation, Communication, Financial Services, and Non Consumption and Non Business Expend Commodities

| Mixture of Transportation, Communication, Financial Services, and Non Consumption and Non Business Expend Commodities |
|-----------------|---|---|---|---|---|---|
| node            | mean | sd  | MC error 2.5% | median | 97.5% | start | sample |
| P7[1]           | 0.2046 | 0.166 | 0.001809 | 0.006371 | 0.1623 | 0.6098 | 501 | 9500 |
| P7[2]           | 0.1993 | 0.1634 | 0.001618 | 0.006569 | 0.1577 | 0.6046 | 501 | 9500 |
| P7[3]           | 0.1979 | 0.162 | 0.001661 | 0.006342 | 0.1585 | 0.6016 | 501 | 9500 |
| P7[4]           | 0.1991 | 0.162 | 0.001639 | 0.006019 | 0.1592 | 0.5957 | 501 | 9500 |
| P7[5]           | 0.1992 | 0.1613 | 0.001635 | 0.006633 | 0.1585 | 0.5949 | 501 | 9500 |
Density Estimation of Kota Mataram Household Expense

<table>
<thead>
<tr>
<th>Mixture of Mixture for Kota Mataram Household Expense</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>node</td>
<td>mean</td>
<td>sd</td>
<td>MC err 2.5%</td>
<td>median</td>
<td>97.5%</td>
<td>start</td>
</tr>
<tr>
<td>phi[1]</td>
<td>0.142</td>
<td>0.1227</td>
<td>0.00118</td>
<td>0.0709</td>
<td>0.1089</td>
<td>S01</td>
</tr>
<tr>
<td>phi[2]</td>
<td>0.1454</td>
<td>0.1264</td>
<td>0.001339</td>
<td>0.0709</td>
<td>0.1098</td>
<td>S01</td>
</tr>
<tr>
<td>phi[3]</td>
<td>0.1409</td>
<td>0.1226</td>
<td>0.001256</td>
<td>0.0708</td>
<td>0.1082</td>
<td>S01</td>
</tr>
<tr>
<td>phi[4]</td>
<td>0.1402</td>
<td>0.1209</td>
<td>0.001237</td>
<td>0.0708</td>
<td>0.1065</td>
<td>S01</td>
</tr>
<tr>
<td>phi[5]</td>
<td>0.1441</td>
<td>0.1236</td>
<td>0.001237</td>
<td>0.0706</td>
<td>0.1065</td>
<td>S01</td>
</tr>
<tr>
<td>phi[6]</td>
<td>0.1439</td>
<td>0.1243</td>
<td>0.001353</td>
<td>0.0706</td>
<td>0.1064</td>
<td>S01</td>
</tr>
<tr>
<td>phi[7]</td>
<td>0.1436</td>
<td>0.1234</td>
<td>0.001235</td>
<td>0.0705</td>
<td>0.1065</td>
<td>S01</td>
</tr>
</tbody>
</table>

Therefore, mixture of mixture for the household expenditure in Kota Mataram is

\[ f(x | \Theta, \pi) = 0.142 \cdot f_1(x | \theta_1, \lambda_1) + 0.1454 \cdot f_2(x | \theta_2, \lambda_2) + 0.1409 \cdot f_3(x | \theta_3, \lambda_3) + \\
0.1402 \cdot f_4(x | \theta_4, \lambda_4) + 0.1441 \cdot f_5(x | \theta_5, \lambda_5) + 0.1439 \cdot f_6(x | \theta_6, \lambda_6) + \\
0.1436 \cdot f_7(x | \theta_7, \lambda_7) \]

Plots of convergence and estimated kernel density of mixture of mixture function of some proportion are described in Figure 1 and Figure 2.

![Convergence Plots of Mixture of Mixture Parameters](image)

The convergence distribution of each proportion are identified to be valid and tend to have the same density. It is supported by testing the equal pattern of some data taken from early iteration after burn-in and some data taken from last iteration of MCMC. The result is shown in Table 2, which has p-value = 0.935 (greater than \( \alpha=0.05 \)).
Table 2. Test for Equal Mean

<table>
<thead>
<tr>
<th>Kolmogorov-Smirnov Z</th>
<th>Asymp. Sig. (2-tailed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.638</td>
<td>0.935</td>
</tr>
</tbody>
</table>

a Grouping Variable: group

Figure 2. Kernel Density Estimation for Proportion of Mixture Of Mixture Model

Conclusion

Based on the information given above we can conclude that the density of Kota Mataram’s household expenditure is formed by 14.2% from Foodstuff Commodities, 14.54% from ready to serve Food, Drink, Cigarette, and Tobacco Commodities, 14.09% from Housing, Electricity, Gas and Power Source Commodities, 14.02% from Clothing Commodities, 14.41% from Health Commodities, 14.39% from Education, Recreation and Sport Commodities, and 14.36% from Transportation, Communication, Financial Services and Non Consumption and Non Business Expend Commodities. It can be seen that the ready to serve Food, Drink, Cigarette, and Tobacco Commodities is the commodity that mostly contribute to the density of Kota Mataram’s household expenditure.
Acknowledgements
This research was sponsored by DIPA Institut Teknologi Sepuluh Nopember (ITS), Surabaya, operated by the Research and Community Services Institute of ITS for the year of 2009. The authors wish to thank to the anonymous referees for their valuable comments which led to considerable improvement in this article and also give chance for presenting this paper in this ICoMS-4 conference.

References
ON CREATING TRUNCATED WEIBULL DISTRIBUTION MODULE IN WINBUGS AND ITS USE IN BAYESIAN FRONTIER FUNCTION MODELING

DEDY DWI PRASYO, NUR IRIAWAN, AND YUNI WULANDARI

Statistics Department
Institut Teknologi Sepuluh Nopember, Surabaya-Indonesia

Abstract. Frontier function model has two structures of error, i.e. error of inefficiency and error of model. Bayesian frontier function model can be estimated using MCMC (Markov Chain Monte Carlo) with auxiliary WinBUGS 1.4. This model has condition, where the distribution of error of inefficiency have to specification be truncated normal or truncated weibull. By looking condition of error of inefficiency, WinBUGS can’t provide facility to analyze two distributions from specification of error. Because of that problem, a new truncated weibull distribution module will be added to WinBUGS 1.4 using HackBox Component Builder software. By this adding new module, bayesian frontier function model with truncated weibull distributions of error of inefficiency can be solved.

Key words and Phrases: Bayesian frontier function, WinBUGS 1.4, Truncated weibull, HackBox component builder

1. INTRODUCTION

Evaluations of efficiency in private company and industry became popular since the development of stochastic frontier production function by [1] and [5]. Frontier function model likes the other linear models, it is built by several input but it has two component of errors, i.e. inefficiency error and error of model. The estimation of frontier function model have to specify the distribution of inefficiency error, truncated normal (half normal) or truncated weibull, at first.

Frontier function correlate with an efficiency measurement and, automatically, relate to the productivity measurement. Frontier parameter can be estimated by using bayesian econometric approach. Bayesian method is a powerful method for estimating parameter, although the available data is small or incomplete. The most popular bayesian algorithm in estimating parameter is MCMC (Markov Chain Monte Carlo).

The computation of MCMC is very complicated. It is needed complex procedures and high skill in computer programming. WinBUGS give facility to estimate model’s parameters by MCMC method. WinBUGS release 1.4 is a bayesian tools which can be found by free in internet. WinBUGS is the current, windows-based, version of the BUGS (Bayesian Inference Using Gibbs Sampler) software. It is a user-friendly, ‘point-and-click’ environment that makes accessible state-of-the-art statistical methodology (MCMC) for analysis of a wide class of bayesian full probability models [4].

In this research, WinBUGS is used to estimate bayesian frontier function model, but WinBUGS 1.4 doesn’t give distribution truncated normal and truncated weibull in its utility (module). This problem cause the bayesian frontier function model can’t be estimated by MCMC algorithm via WinBUGS. Lunn and Jackson (2004) give an interesting help for creative WinBUGS user in developing their own distribution. This guidance very useful for user who has
specific problems with particular pattern of data which does not supported by WinBUGS utilities. This paper shows how to create truncated weibull distribution in WinBUGS 1.4 by using Black-Box Component Builder and apply this new module in bayesian frontier function modeling.

2. BAYESIAN FRONTIER FUNCTION MODELING

Stochastic frontier is one of the popular method used in productivity analysis. This method is usually used for calculating the efficiency of a factory, i.e. production, cost, or profit analysis. Frontier function model is built by several input and two stochastic component of errors, i.e. error of model and error of inefficiency. Error of model is a noise factor that can’t be controlled. The second error is inefficiency error that could be controlled by factory. The distribution of error of inefficiency is usually truncated normal (half normal) or truncated weibull.

Productivity is a comparison of output and input. Two important aspects in productivity analysis are efficiency and effectivity. The efficiency could be measured by how well all the input could be combined. The effectivity is measured by how the output could reach the target. The productivity could be measured by several methods, i.e. index, production function, or input-output approaches. This research focuses on production function approach, especially the stochastic frontier production function approach.

Aigner, Lovell, and Schmidt (1977) and Meuwen and Van Den Broeck (1977) said that stochastic frontier function is as in the following equation:

\[ y_i = f(x_i; \beta) + \varepsilon_i \]  

where

\[ \varepsilon_i = v_i - |x_i| \]

with \( y \) is dependent variable or output, \( x \) is dependent variable or input, \( \beta \) is vector of coefficient parameter, and \( \varepsilon \) is error. Variable \( v \) is error of model and \( u \) is error of inefficiency. Error of model is normally distributed or \( v_i \sim IIDN(0, \sigma^2) \) and \( u \) is usually truncated (positive half) normally distributed or \( u_i \sim IIDN(0, \sigma^2) \). Error of inefficiency can be also exponentially distributed or truncated normal with non zero mean.

Production function usually used in stochastic frontier function is Cobb Douglas Production Frontier which is stated as in the following:

\[ y_i = e^a \prod_{j=1}^{p} x_{ij}^{\beta_j} e^{\varepsilon_i}, \quad i = 1, 2, ..., n \]  

where \( y \) is output variable, \( x \) is input variable, \( \beta \) is coefficient parameter of input variable, \( v \) and \( u \) are error of model and error of inefficiency, and \( n \) is the number of data. The economic analysis is usually emphasize on elasticity of coefficient \( \beta \). Cobb Douglas function in eq. (2) can be transformed in to the following equation

\[ \ln y_i = \alpha + \beta_1 \ln x_{i1} + ... + \beta_p \ln x_{ip} + v_i - u_i \]  

Bayesian statistics differ from the statistical theory since all unknown parameters are considered as random variables [6]. For this reason, prior distribution must be defined initially. This prior distribution expresses the information available to the researcher before any data are involved in the statistical analysis. According to the Bayes theorem, the posterior distribution can be written as

\[ f(\theta|y) = \frac{f(y|\theta) f(\theta)}{f(y)} \propto f(y|\theta) f(\theta) \]  

2
where $\theta$ is vector parameter involve coefficient parameters of input variables and errors, $f(\theta)$ is prior distribution, and

$$f(y|\theta) = \prod_{i=1}^{n} f(y_{i}|\theta)$$

is joint distribution which is called likelihood of the model.

3. TRUNCATED WEIBULL DISTRIBUTION

If $U$ is variable random having distribution truncated weibull with $c$ is shape parameter, $b$ is scale parameter, and $t$ is left truncation value, so the probability distribution function (pdf) and its cumulative (CDF) of Left Truncated Weibull (LTW) distribution is as in the following [8]:

$$f(u|c,b) = \frac{c}{b} \left( \frac{u}{b} \right)^{c-1} \exp \left( -\frac{u}{b} \right) + \frac{1}{b}$$

(6)

$$F(u) = 1 - \exp \left( -\frac{u}{b} \right) + \frac{t}{b}$$

(7)

where $b > 0$, $c > 0$, and $u \geq t$, then likelihood function of LTW distribution in eq. (6) can be stated as in the following equation:

$$L(c,b) = \prod_{i=1}^{n} f(u_{i})$$

$$L(c,b) = \left( \frac{c}{b} \right)^n \frac{1}{b^{c-1}} \exp \left( -\sum_{i=1}^{n} \frac{u_{i}}{b} \right) + \frac{t}{b}$$

(8)

LTW distribution with different truncation value, shape and scale parameter and can be showed in Fig. 1 as follow.

![Figure 1. LTW Distribution](image)

4. MCMC METHOD USING GIBBS SAMPLING

One of the attractive methods for setting up an MCMC algorithm is Gibbs sampling. Suppose that the parameter vector of interest is $\theta = (\theta_1, \theta_2, ..., \theta_n)$. The joint posterior distribution of $\theta$, which we denote by $(\theta|data)$, may be of high dimension and difficult to summarizes. Suppose we
define the set of conditional distributions

\[
\begin{align*}
\theta_1 | \theta_2, \theta_3, ..., \theta_p, \text{data} \\
\theta_2 | \theta_1, \theta_3, ..., \theta_p, \text{data} \\
& \quad \vdots \\
\theta_p | \theta_1, \theta_2, ..., \theta_{p-1}, \text{data}
\end{align*}
\]

where \( p \) is the number of parameter and \( \theta_1 | \theta_2, \theta_3, ..., \theta_p, \text{data} \) represents the distribution of \( \theta_1 \) conditional on values of the random variables \( \theta_2, \theta_3, ..., \theta_p \), and \text{data}. The idea behind Gibbs sampling is that we can set up a markov chain simulation algorithm from the joint posterior distribution by successfully simulating individual parameters from the set of \( p \) conditional distributions. Simulating one value of each individual parameter from these distribution is turn called one cycle of Gibbs sampling. Under general condition, draws from this simulation algorithm will converge to the target distribution (the joint posterior of \( \theta \)) of interest [2].

In situation where it is not convenient to sample directly from the conditional distribution, one can use a Metropolis algorithm such as the random walk type to simulate from each distribution. Suppose that \( \theta_i' \) represent the current value of \( \theta_i \) in the simulation, and let \( g(\theta_i) \) represent the conditional distribution where we have suppressed the dependence of this distribution on values of remaining components of \( \theta \). Then the candidate value for \( \theta_i \) is given by

\[
\theta_i^* = \theta_i' + c_i Z
\]

where \( Z \) is standard normal variate and \( c_i \) is a fixed scale parameter. The next simulated value of \( \theta_i \), that is \( \theta_i^{* \star} \), will be equal to the candidate value with probability [2]:

\[
Pr = \min \left( 1, \frac{g(\theta_i^*)}{g(\theta_i')} \right)
\]

otherwise the value \( \theta_i^{* \star} = \theta_i' \).

5. MAIN RESULTS

The data used in this research is electricity production per-kwh as dependent variable, electricity consumption per-kwh per-type of tariff, and percentage of losses as error of inefficiency. Type of tariff is divided into social (S), housing (H), business (B), industry (I), government (G), and multiuse (M). Losses are the difference of electricity production and electricity consumption divided by the electricity production. Data are collected from January 2004 until December 2008 in one city in East Java, Indonesia.

The consumption of electricity can be ordered from the highest to the lowest as follow: housing, industry, business, government, social, and multiuse. The percentage of losses relatively decreased as shown in time series plot in Fig. (2a). The frontier function is usually uses panel data, so the effect of time is neglected. The histogram as in Fig. (2b) shows that the distribution of percentage of losses is not symmetric, but it skew distributed. The distribution of error inefficiency is usually truncated normal or truncated weibull.
The Goodness of Fit (GoF) test is done for determining the distribution of percentage of losses. Kolmogorov-Smirnov (KS) test is very popular for GoF test. KS tests show that percentage of losses has truncated normal or truncated weibull distribution, but truncated weibull give more powerful result. Because of this reason, this research uses truncated weibull as the distribution of error of inefficiency. The new problem coming is the truncated weibull distribution module is not available in WinBUGS utility.

5.1. Creating Truncated Weibull Distribution in WinBUGS

The Black-Box Component Builder provide a user interface that enables rapid application development: it incorporates a powerful text editor, for example, and to some extent automates the programming of many of the essential features of modern software. WinBUGS has been designed so that users can extend it, with minimal effort, to meet their own requirements. All that is required is some familiarity with the Component Pascal language and structure of the WinBUGS framework [4].

Steps of adding new module in WinBUGS is as follow [7]:

- Download BlackBox Component Builder from http://www.oberon.ch/blackbox.html
- Unzip BlackBox file by clicking icon SetupBlackBox15.exe (put in C:\Program Files/BlackBox)
- Copy (Ctrl+C) all files in directory of WinBUGS14 (Program Files/WinBUGS14) and Paste (Ctrl+V) them to BlackBox directory (Program Files/BlackBox). Choose “Yes to All” to replace all files already in BlackBox directory.
- Now WinBUGS 14 and BlackBox are ready to be used naturally for Bayesian Modeling and developing modules.
- Download additional component wbdev_shared_source.exe and wbdev.exe from http://www.winbugs-development.org.uk and the extract those files to Program Files\BlackBox Component Builder 1.5\.
- Open file: Program Files\BlackBox Component Builder 1.5\wbdev_shared_coded.txt using BlackBox, then decode the file. Do the same way to file wbdev_01_09_04.txt. Close Black-Box and open again to develop WinBUGS.

A new module can be developed by modifying Univariate Template module (see in Program Files/BlackBox/WBDev/Mod/UnivariateTemplate.odc). The new truncated weibull distribution module is defined in MODULE
WBDevWeibullTruncLeft; and successfully validated. Truncated weibull distribution had successfully added to the utility in WinBUGS and be ready to be used in bayesian frontier function modeling.

5.2. Bayesian Frontier Function Modeling

The bayesian frontier function modeling is implemented in electricity data previously described via open source WinBUGS 1.4. The frontier function used in this research is Cobb Douglas function in Eq. (3) with error of inefficiency is truncated weibull distributed. The doodle in WinBUGS and the source code for this modeling are shown in Fig. (3) and Fig. (4).

Figure 3. Doodle for Cobb Douglas Function with Error of Inefficiency is Truncated Weibull Distributed

Figure 4. Source Code of Doodle in Fig. 3.
Table 1. Parameter Estimation using MCMC

<table>
<thead>
<tr>
<th>node</th>
<th>mean</th>
<th>sd</th>
<th>2.5%</th>
<th>median</th>
<th>97.5%</th>
</tr>
</thead>
<tbody>
<tr>
<td>alpha</td>
<td>3.90</td>
<td>1.06</td>
<td>1.83</td>
<td>3.899</td>
<td>5.98</td>
</tr>
<tr>
<td>beta[1] or S</td>
<td>-20.20</td>
<td>1.31</td>
<td>-22.76</td>
<td>-20.19</td>
<td>-17.62</td>
</tr>
<tr>
<td>beta[3] or B</td>
<td>22.88</td>
<td>1.34</td>
<td>20.26</td>
<td>22.88</td>
<td>25.51</td>
</tr>
<tr>
<td>beta[4] or I</td>
<td>24.33</td>
<td>1.54</td>
<td>21.30</td>
<td>24.33</td>
<td>27.35</td>
</tr>
<tr>
<td>beta[6] or M</td>
<td>0.08</td>
<td>0.55</td>
<td>-0.99</td>
<td>0.08</td>
<td>1.17</td>
</tr>
<tr>
<td>Truncated</td>
<td>0.00559</td>
<td>0.003207</td>
<td>2.80E-01</td>
<td>0.005597</td>
<td>0.01086</td>
</tr>
<tr>
<td>sigma</td>
<td>1.215</td>
<td>0.279</td>
<td>0.77</td>
<td>1.181</td>
<td>1.857</td>
</tr>
</tbody>
</table>

Parameters estimation of Cobb Douglas function by bayesian inference is done through MCMC method using Gibbs sampling which became the default of bayesian computation in WinBUGS. Mean Square Error (MSE) of the model is represented by the mean value of variable sigma. Point estimation of the coefficient parameters is the mean value of cycle in MCMC. Standard deviation for every node represent the standard error of every coefficient parameter in the model. Bayesian frontier function for production and consumption electricity data with error of inefficiency is truncated weibullly distributed is stated as in the following equation:

\[
\ln y_i = 3.9 - 20.2 S - 15.34 H + 22.88 B + 24.33 I + \\
-12.59 G + 0.08 M + y_i - u_i \quad (11.a)
\]

or

\[
y_i = e^{3.9 S - 20.2 H - 15.34 B - 22.88 I - 24.33 G - 12.59 M + 0.08 e - u_i} \quad (11.b)
\]

Model in Eq. (11) show that the one percent additional of input kwh from social tariff will reduce 20% of the electricity production (in ln value). In the other hand, one percent additional of input from industry sector will increase the electricity in amount 24.33%.

6. CONCLUDING REMARKS

Truncated weibull distribution that has succeed to be implemented as an add-ins in WinBUGS 1.4 is possible to be used for bayesian frontier function modeling. Bayesian inference is relatively improve the maximum likelihood in estimating frontier function model. The electricity consumption in industry is the highest contributor to increase the electricity production.

Acknowledgement. This research was sponsored by DIPA Institut Teknologi Sepuluh Nopember (ITS), Surabaya, operated by the Research and Community Services Institute of ITS for the year of 2009. The author wishes to thank the anonymous referees for their valuable comments which led to considerable improvement in this article.

REFERENCES


Dedy Dwi Prastyo. Department of Statistics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia. e-mail: dechyp@statistika.its.ac.id

Nur Iriawan. Department of Statistics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia. e-mail: nur_i@statistika.its.ac.id