An extended logit-normal regression with application to human development index data

Submitted by

Bipplav Kumar Tiwari (200281)

Siddhant Shivdutt Singh (200977)

Tanush Kumar (201043)

Vineet Kumar (201121)

Supervised by

Prof. Arnab Hazra

DEPARTMENT OF MATHEMATICS AND STATISTICS

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

November 2022

Abstract

In several real world problems, we need to model data with a variety of shapes such as data having two peaks, or bimodal data. We often have to resort to mixture distributions in order to model them. The paper [1] proposes a single distribution which can account for the bimodality of our data. In this report, we have established some theoretical properties and practical applications of the proposed $OLLLTN$ distribution. We began by defining its PDF, CDF, moments, skewness, and kurtosis, and then illustrated some of them using plots. We went ahead and implemented functions to sample from the distribution and return values of the above properties for the distribution. We then described $OLLLTN$ regression and tested the same on the dataset we generated using our sampler. Finally we modelled the Human Development Index (HDI) data of two Brazilian cities using $OLLLTN$ distribution and compared the quality of fit with other distributions to establish its practicality.

We were able to reproduce most of the results and findings that are proposed in the paper and implement the same in R. The pdf of our presentation slides and the Brazilian HDI dataset is uploaded here.

Contents

Chapter 1

Introduction

1.1 Background

In the past years, the *logit-normal* (LTN) distribution has proved to be very useful in analyzing data in the interval of (0,1). It is specially useful in areas of manufacturing and delivery and is used to analyze the data of manufacturing and delivery times. It is also used in Engineering to model the lifetime of systems whose failure modes involve fatigue stress. LTN can also have some other very interesting applications like, a linear mixed model based on LTN can be used to model the nature and variability of rainfall during the Indian monsoons. LTN distribution has been researched extensively as a generator family, i.e it is used to generate new probability distributions.

1.2 Motivation

In several real world applications, we are faced with situations where we want to be able to generate predictions about the frequency or magnitude of a phenomenon, but all we have is past data. In these scenarios, data fitting using a probability distributions can be a useful. For example, we commonly use the normal distribution to fit symmetric data centered around the mean. These distributions can have one peak, or they can have several peaks. The probability distribution that best fits them varies according to the shape of the distribution of data. The familiar normal distribution, or bell curve, has one peak, or unimodal. Similarly, we define distributions with two peaks as bimodal distributions.

Generally a mixture of distributions has to be used to model the bimodal proportional

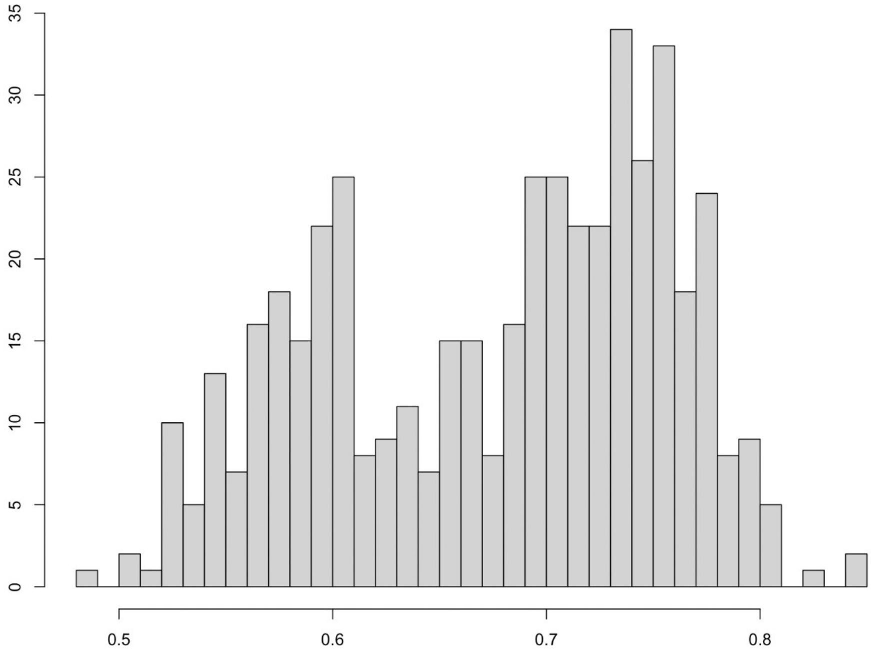

FIGURE 1.1: The HDI (Human Development Index) histogram for 478 cities in the Brazilian states of Santa Catarina and Pernambuco. The formation of two sub-populations can be clearly noted, so these data have a bimodal shape.

data. In this paper the author has proposed the odd log-logistic logit-normal (OLLLTN) distribution to model bimodal proportional data without the need of a mixture of distributions. As the name suggests, The distribution is generated using the odd log-logistic generator (OLL-G) family on the logit-normal (LTN) distribution. The logit-normal (LTN) distribution is very useful to analyze data in the interval (0, 1). It is commonly used to generate other probability distributions. The author argues that the OLLLTN distribution can be an interesting alternative to bimodal data in the (0, 1) interval instead of the classical beta and simplex models.

Chapter 2

Introduction to OLLLTN

2.1 Derivation

Lets start by defining LTN distribution. If $W \sim Normal(logit(\mu)), \sigma^2)$, where $logit(\mu) = log(\mu/1-\mu)$ and $\mu \in (0,1)$, The LTN random variable is defined by $X = (1+e^{-W})^{-1}$. The cdf of X (for $0 < x < 1$) is:

$$G_{\mu,\sigma}(x) = \psi(\frac{logit(x) - logit(\mu)}{\sigma}) = \frac{1}{2}[1 + erf(\frac{logit(x) - logit(\mu)}{\sqrt{2}\sigma})] \tag{2.1}$$

where $\psi(z)$ is the standard normal cdf and $erf(z) = 2\pi^{\frac{1}{2}} \int_0^z e^{\frac{-t^2}{2}}$ is the error function. Now, from cdf we get the pdf of X to be

$$g_{\mu,\sigma}(x) = \frac{1}{x(1-x)\sqrt{2\pi\sigma^2}} e^{\frac{-[logit(x)-logit(\mu)]^2}{2\sigma^2}} \tag{2.2}$$

The odd *log-logistic generator* (OLL-G) family follows by integrating the log-logistic density function. The density of a log-logistic distribution is:

$$f(x;\alpha,\beta) = \frac{(\beta/\alpha)(x/\alpha)^{\beta-1}}{(1+(x/\alpha)^{\beta})^2} \tag{2.3}$$

where $\alpha > 0$ is a scale parameter and also the median of the distribution, $\beta > 0$ is a shape parameter. The distribution is unimodal when $\beta > 1$ and its dispersion decreases as β increases.

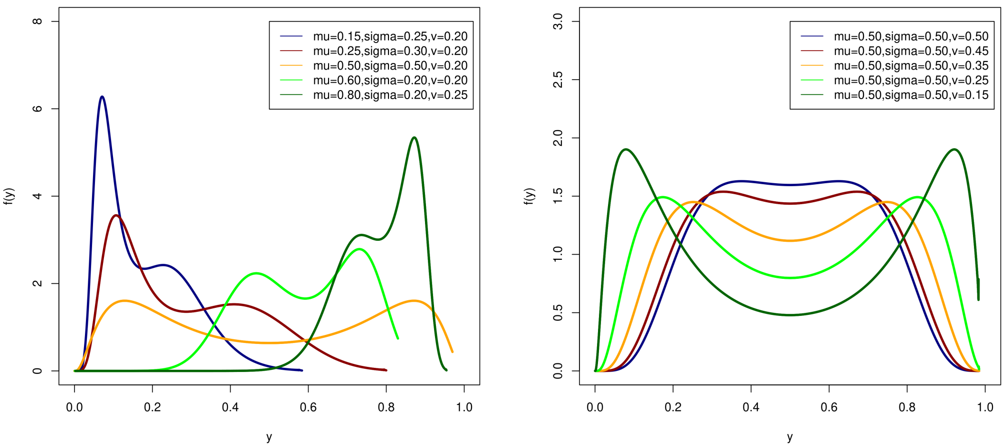

FIGURE 2.1: Density of OLLLTN distribution. The above plots were generated using R.

2.2 Cumulative Distribution Function (cdf)

As said earlier, The odd *log-logistic generator* (OLL-G) family is derived by integrating the log-logistic density function. Hence, The cdf of OLLTN is given by:

$$F(y) = F(y;\mu,\sigma,\nu) = \int_0^{\frac{G_{\mu,\sigma(y)}}{1-G_{\mu,\sigma(y)}}} \frac{\nu x^{\nu-1}}{(1+x^{\nu})^2} dx = G_{\mu,\sigma}(y)^{\nu} \tag{2.4}$$

where $0 < \mu < 1$ is a position, $\sigma > 0$ is a scale, and $\nu > 0$ is a shape parameter.

2.3 Probability Density Function (pdf

We can derive the pdf of OLLLTN distribution by simply differentiating the cdf. For simplicity of representation, we are assuming $\eta(y) = G_{\mu,\sigma}(y)$. Then, the pdf of OLLLTN distribution can be written as:

$$\begin{aligned} f(y) = f(y;\mu,\sigma,\nu) &= \frac{\nu}{y(1-y)\sqrt{2\pi\sigma^2}} e^{\frac{-[logit(x)-logit(\mu)]^2}{2\sigma^2}} \\ &\times [\eta(y)[1-\eta(y)]]^{\nu-1}[\eta(y)^{\nu} + [1-\eta(y)]]^{\nu-2} \end{aligned} \tag{2.5}$$

It can be clearly seen that the LTN distribution is a special case of the OLLLTN model when $\nu = 1$. Density plots of Y in Figure 2.1 shows bimodality (for $0 < \nu < 1$), whereas the LTN model does not have bimodality.

Chapter 3

Properties of OLLLTN

By inverting Eq.(2.1), we obtain the quantile function (qf) of the LTN distribution $Q_{LTN}(u) = G^{-1}_{\mu,\sigma}(u) = 1 + exp[-v(u)]^{-1}$, where $v(u) = logit(\mu) + \sqrt{2}\sigma erf^{-1}(2u-1)$ depends on the inverse error function. This inverse can be computed from $erf^{-1}(z) = \sqrt{\pi}(z/2 + \pi z^3/24 + 7\pi^2 z^5/960 + ...)$ but $Q_{LTN}(u)$ is easily calculated in the gamlss package in R.

3.1 Moments

The nth ordinary moment of the OLLLTN model can be computed numerically from

$$k'_n = E(Y^n) = \int_0^1 Q_{LTN}\left(\frac{u^{1/v}}{u^{1/v} + [1-u]^{1/v}}\right)^n du \tag{3.1}$$

The first four moments of Y calculated in R software from (3.1) are reported in Table 1. The nth incomplete moment of Y is given by

$$m_n(q) = E(Y^n|Y \leq p) = \int_0^{F(p)} Q_{LTN}\left(\frac{u^{1/v}}{u^{1/v} + [1-u]^{1/v}}\right)^n du \tag{3.2}$$

Where $F(p) = Q_{LTN}\left(p^{1/v} p^{1/v} + [1-p]^{1/v}\right)$. This equation with n=1 is useful to determine the mean deviations and Bonferroni and Lorenz curves of Y.

TABLE 3.1: Values for $\kappa_1, \kappa_2, \kappa_3 and \kappa_4$.

μ	σ	ν	κ_1	κ_2	κ_3	κ_4
0.2	0.1	0.1	0.22532054	0.07068196	0.03576166	0.02706187
0.2	0.1	2	0.200141025	0.040131762	0.008062229	0.001622591
0.5	0.8	0.1	0.5000172	0.4329052	0.3993577	0.3758595
0.5	2	0.1	0.5	0.4781088	0.4671633	0.4596786
0.8	0.5	0.1	0.6579184	0.5436995	0.4856778	0.4484023
0.8	0.1	2	0.799859	0.6398497	0.5119104	0.4096002

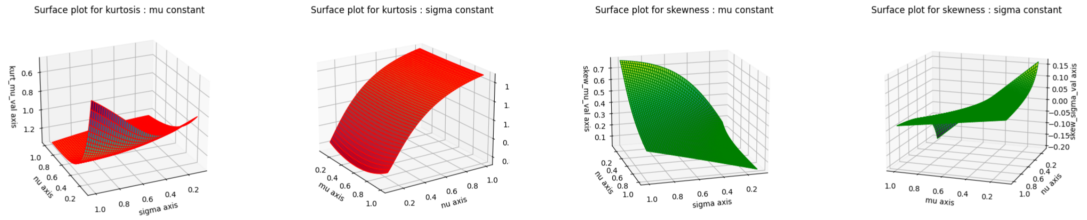

FIGURE 3.1: (a) and (b) Moors kurtosis. (c) and (d) Bowley's skewness.

3.2 Skewness and Kurtosis

The effects of the parameters of the OLLLTN distribution can also be measured by the skewness and kurtosis computed from its qf $Q_Y(.)$. The Bowley's skewness and Moors' kurtosis are less sensitivity to outliers and given by

$$Skewness = \frac{Q_Y(1/4) + Q_Y(3/4) - 2Q_Y(1/2)}{Q_Y(3/4) - Q_Y(1/4)} \tag{3.3}$$

$$Kurtosis = \frac{Q_Y(7/8) - Q_Y(5/8) + Q_Y(3/8) - Q_Y(1/8)}{Q_Y(6/8) - Q_Y(2/8)} \tag{3.4}$$

respectively. Moreover, they are alternative forms for the measures based on moments for the OLLLTN distribution since the last ones do not have explicit expressions. Plots of Bowley's skewness (3.3) and Moors' kurtosis (3.4) of the OLLLTN model are reported in Figure 4, which reveal how they depend on the parameter values. In Figure 3.1(a) $\mu = 0.1$ is fixed, and the other parameters vary $\sigma \in [0.1, 1]$ and $\nu \in [0.1, 1]$.In Figure 3.1(b) $\sigma = 0.1$,$\mu \in [0.1, 1]$ and $\nu \in [0.1, 1]$. In Figure 3.1(c) $\mu = 0.1$,$\sigma \in [0.1, 1]$ and $\nu \in [0.1, 1]$. In Figure 3.1(d) $\sigma = 0.1$,$\mu \in [0.1, 1]$ and $\nu \in [0.1, 1]$. The script for building these plots is given in the Appendix.

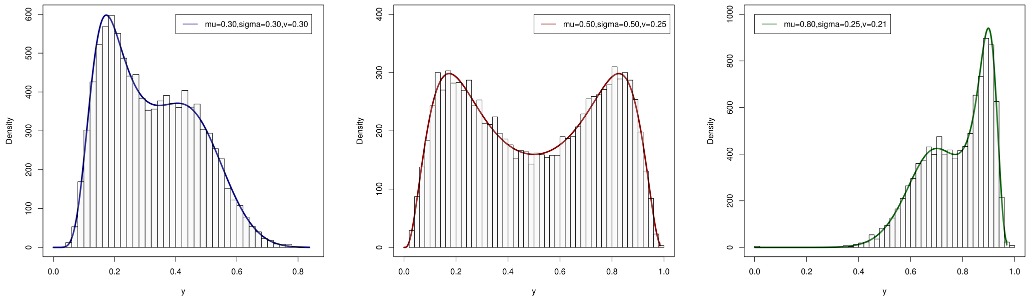

FIGURE 3.2: Histograms and plots of OLLTN density

3.3 Sampling from OLLLTN

The quantile function of Y has a simple form

$$y = Q_Y(u) = Q_{LTN}\left(\frac{u^{1/v}}{u^{1/v} + [1-u]^{1/v}}\right) \tag{3.5}$$

To sample from the distribution, we have used the inverse transform method since $F^{-1}(u) = Q_Y(u)$ was easily available to us through the gamlss package. We created a vector of length 5,000 and then run a loop 5,000 times and each time we run the loop, we sample a number from uniform(0,1), calculate Y from it, and assign it in the vector. Finally, we use the *hist*() function to plot the histograms that are mentioned in the figure 3.2 for three different configurations to obtain three different plots representing different types of bimodality that the OLLLTN distribution can model.

Chapter 4

OLLLTN Regression

4.1 LTN Regression

The LTN regression for proportional data is constructed by liking the parameters μ_i and σ_i of the response Y_i to the vector ${x_i}^\top = (x_{i1}, ...x_{ip})$ of covariates, (for our case the number of dimensions i.e p is equal to 3)

$$\mu_i = \frac{exp(x_i^\top \beta_1)}{1 + exp(x_i^\top \beta_1)} \quad and \quad \sigma_i = exp(x_i^\top \beta_2), i = 1, ..., n, \tag{4.1}$$

respectively where $\beta_1 = (\beta_{11}, ..., \beta_{1p})$ and $\beta_2 = (\beta_{21}, ..., \beta_{2p})$ are unknown coefficients. We used the values obtained through LTN regression as the starting point for OLLLTN regression. For this purpose, the log-likelihood equation is

$$l(\theta) = \sum_{i=1}^{n} log\left[\frac{1}{y_i(1-y_i)}\right] - \frac{n}{2}log(2\pi) - \frac{1}{2}\sum_{i=1}^{n} log(\sigma_i^2) - \frac{1}{2}\sum_{i=1}^{n}\left\{\frac{[logit(y_i) - logit(\mu_i)]^2}{\sigma_i^2}\right\} \tag{4.2}$$

4.2 OLLLTN Regression

The log-likelihood function for $\theta = (\beta_1^\top, \beta_2^\top, \beta_3)^\top$ for regression (11) given n independent observations is

$$\begin{aligned} l(\theta) = {} & nlog(\nu) + \sum_{i=1}^{n} log\left[\frac{1}{y_i(1-y_i)}\right] - \frac{n}{2}log(2\pi) - \frac{1}{2}\sum_{i=1}^{n} log(\sigma_i^2) \\ & - \frac{1}{2}\sum_{i=1}^{n}\left\{\frac{[logit(y_i) - logit(\mu_i)]^2}{\sigma_i^2}\right\} + (\nu - 1)\sum_{i=1}^{2} log\{\eta(y_i)[1 - \eta(y_i)]\} \\ & - 2\sum_{i=1}^{n} log\{\eta(y_i)^\nu + [1 - \eta(y_i)]^\nu\}. \end{aligned} \tag{4.3}$$

By maximizing (4.3) numerically in the optim package, it follows the maximum likelihood estimate (MLE) $\hat{\theta}$. Initial values for β_1 and β_2 were obtained by fitting the given data to the LTN regression model ($\nu = 1$) and some corrections were also made to the setting of the original paper such as the use of β_3 as the parameter instead of using ν directly. This helped us to code the whole process using just optim function of the R package since we extended the domain of the third parameter (previously $\nu \in (0, 1)$) to the real number line. We did this by setting $\nu = \frac{exp(\beta_3)}{1+exp(\beta_3)}$ i.e using a logit function to represent ν in terms of β_3. We are getting very accurate measurements as shown in Sec. 5, where some simulations are reported under different scenarios.

Chapter 5

Monte Carlo Simulation

After establishing the theoretical properties of the OLLLTN distribution, we now aim to conduct some simulation studies to test the practicality of the proposed distribution. The first Monte Carlo experiment is described in this section.

5.1 Experimental Set Up

We want to test whether the OLLLTN distribution can be used to fit bimodal data. To conduct this experiment, we first generate simulated bimodal data and then try to find the OLLLTN distribution parameters that best fit it.

1. For j in $[1, n]$
2. Set μ, σ, ν with $\mu \in (0,1)$, $\sigma > 0$, $\nu \in (0,1)$
3. Generate $Y_i \sim OLLLTN(\mu, \sigma, \nu)$ for $i = 5000$. We generated these samples using the technique described in the previous section.
4. Now that we have simulated bimodal data, we want to fit the OLLLTN distribution on this distribution. Thus we generate predictions $\hat{\mu_j}, \hat{\sigma_j}, \hat{\nu_j}$ by minimizing the negative log likelihood function defined in the previous section.
5. Calculate B_j (Bias), and M_j for each parameter (Mean square error) defined as following

$$B_j = \mu - \hat{\mu_j}$$

$$M_j = (\mu - \hat{\mu_j})^2$$

Similarly it is defined for other parameters σ and ν

6. Calculate Average Estimate, Average Bias and Average Mean Square Error

7. Repeat the above experiment for n=100,500 and 1000 for each of the following scenarios:

$$Scenario1 : \mu = 0.3, \sigma = 0.3, \nu = 0.3$$

$$Scenario2 : \mu = 0.5, \sigma = 0.5, \nu = 0.25$$

$$Scenario3 : \mu = 0.8, \sigma = 0.25, \nu = 0.21$$

5.2 Results

We conducted the above exeriment in R. Samples were generated as described in the previous section. For negative log likelihood minimization, we first wrote a function in R that takes the three parameters as input and returns the value of the negative log likelihood value. We used the *"L-BFGS"* method in optim to obtain the estimates. The results were tabulated in the table below. They were in accordance to the results in the original paper.

Table 5.1: Results from the simulation study described in the above section

Scenario	Parameter	$n = 100$			$n = 500$			$n = 1000$		
		AE	Bias	MSE	AE	Bias	MSE	AE	Bias	MSE
1	μ	0.300	0.000	0.004	0.300	-0.000	0.005	0.300	0.000	0.004
	σ	0.297	0.003	0.030	0.300	0.000	0.026	0.302	-0.002	0.059
	ν	0.902	-0.002	0.085	0.905	-0.005	0.076	0.902	-0.002	0.066
2	μ	0.499	0.001	0.010	0.499	0.001	0.011	0.499	0.001	0.008
	σ	0.532	-0.032	0.170	0.528	-0.028	0.164	0.527	-0.027	0.112
	ν	0.268	-0.018	0.131	0.272	-0.022	0.142	0.264	-0.014	0.086
3	μ	0.800	0.000	0.003	0.800	0.000	0.004	0.800	0.000	0.003
	σ	0.326	-0.076	0.266	0.319	0.069	0.268	0.309	-0.059	0.169
	ν	0.308	-0.098	0.363	0.304	-0.094	0.362	0.291	-0.081	0.234

5.3 Remarks and Conclusions

- The author suggests performing parameter estimation using the *"gamlss"* package in R, but we used *optim* as it was syntactically more convinient.

- We used *logit, log* and *logit* for μ, σ and ν respectively as link functions inside the *optim* function to allow the function to take values in the entire real domain and thus ensure convergence.

- The general trend that is observed in the paper is observed here as well. As n increases, our estimates converge to the true estimates, while bias and MSE decrease.
- For lower sigma and nu, the accuracy was not quite as much due to precision errors while calculating the erf function.
- The experiment establishes that given bimodal data, we can use the *OLLLTN* distribution to fit that data.

Chapter 6

OLLLTN Regression Simulation

In the second experiment, we want to simulate regression using the *OLLLTN* distribution, to propose that we can use this distribution to quantify the correlation between a bimodal data parameter and a set of covariates. The theory behind the *OLLLTN* regression was described in section 4.

6.1 Experimental Set Up

We generate synthetic data with a fixed set of covariate coeffecients and then conduct regression on that data. For the purpose of this experiment, we use two sets of covariates and six coeffecients.

1. For j in $[1, n]$
2. Set $\beta_{10} = 0.7, \beta_{11} = 0.1, \beta_{12} = 0.9, \beta_{20} = 0.3, \beta_{21} = 0.5, \beta_{22} = 0.2, \nu = 0.3$
3. Sample $X_{i1} \sim Normal(0, 1), X_{i2} \sim Binomial(1, 0.5)$ for $i = 1, \ldots, 5000$.
4. $\mu_i = \frac{exp(\beta_{10}+\beta_{11}x_{i1}+\beta_{12}x_{i2}}{1+exp(\beta_{10}+\beta_{11}x_{i1}+\beta_{12}x_{i2})}$ and $\sigma_i = exp(\beta_{20} + \beta_{21}x_{i1} + \beta_{22}x_{i2})$.
5. Sample $Y_i \sim OLLLTN(\mu_i, \sigma_i, \nu_i)$.
6. Conduct *LTN* Regression with starting values 0.5 for each parameter to get an initial estimate for β_1, β_2
7. Now conduct *OLLLTN* Regression with starting values $(\beta_1, \beta 2, 1)$ to get the final estimates, $\hat{\beta_{ij}}$ and $\hat{\nu}$
8. Calculate the Bias and Mean Square Error for each parameter
9. Using the estimates from each iteration, calculate the Average Bias and Average Mean Square Error

10. Conduct the above experiment for $n = 100, 200, 500$

6.2 Results

The author suggests using *"gamlss"* here as well, but we use the *optim* function here as well. To do this, we defined a function that takes β_{ij} and ν as input and outputs the negative log likelihood. We use the *"L-BGFS"* method. The results are tabulated below:

Table 6.1: Results from the *OLLLTN* Regression on simulated data

Parameter	$n = 100$			$n = 200$			$n = 500$		
	AE	Bias	MSE	AE	Bias	MSE	AE	Bias	MSE
β_{10}	0.691	0.009	0.065	0.699	0.001	0.059	0.699	0.001	0.056
β_{11}	0.093	0.007	0.004	0.097	0.002	0.003	0.098	0.002	0.003
β_{12}	0.902	-0.002	0.085	0.905	-0.005	0.076	0.902	-0.002	0.066
β_{20}	0.325	-0.025	0.096	0.307	-0.007	0.070	0.319	-0.019	0.104
β_{21}	0.497	0.003	0.009	0.499	0.001	0.007	0.499	0.001	0.010
β_{22}	0.199	0.001	0.015	0.201	-0.001	0.017	0.201	-0.001	0.013
ν	0.315	-0.015	0.057	0.305	-0.005	0.037	0.302	-0.002	0.025

6.3 Remarks and Conclusions

- Initially, we did not define a link function for nu and thus the optimization was highly unstable. But upon adding a logit link function for nu, the optimization became very stable and convergence occured in maximum cases to the right value.
- Convergence is observed in about 90% of the cases
- The trend observed in the paper is observed here as well. For larger n, the average estimates converge to their true values, while MSE and Bias decrease.
- In the paper the above experiment was conducted for $n = 100, 500, 1000$ but to save time, we conducted it for $n = 100, 200, 500$

Chapter 7

Applications

We compared the OLLLTN, LTN, beta (BE) and simplex regressions using the Human Development Index (HDI) data of 478 municipalities in the Brazilian states of Pernambuco and Santa Catarina. The MLEs, the global deviance (GD) and the Akaike Information Criterion(AIC) were calculated using the $GAMLSS$ package in R software.

- We adopted the BE density (implemented in GAMLSS) with mean $0 < \mu < 1$ and dispersion $0 < \sigma < 1$, where the variance is $\sigma^2\mu(1-\mu)$.
- We considered the simplex density (Barndorff-Nielsen and Jørgensen 1991) with mean $\mu \in (0,1)$ and variance $\sigma^2\mu^3(1-\mu)^3$, and then $\sigma^2 > 0$ is the dispersion.

7.1 Brazil HDI Data

The HDI is a metric used to ascertain the level of development and associated quality of life of a determined population, involving education, health and income. This indicator was formulated by the United Nations Development Program (UNDP) in 1990 as a broader measure of welfare than just economic questions. It is measured by the geometric mean of the normalized indices of these three dimensions of human development. The first is the education component given the average number of years of formal education of a population. The second is the health component given by the average life expectancy of the population of interest. The assumption is that longer life expectancy is associated with better quality of life. Finally, the third is the income component based on the gross domestic product (GDP) per capita. In this study, we are basically interested in the HDI

of the states of Santa Catarina and Pernambuco, which are geographically distant: Santa Catarina is located in Brazil's South region while Pernambuco is in the Northeast.
We analyze the data marginally by fitting four distributions to the HDI observations. The estimates and the adequacy measures are given in 7.1. The smallest criteria correspond to the OLLLTN distribution which gives the best fit to these data. 7.1 displays the histogram and the density plots, whereas Figure 6(b) provides plots of the empirical and estimated cdfs. We conclude that the OLLLTN distribution is the best model for these data and the only one to cope with bimodality.

7.2 Results

We fit four distributions to the Brazilian HDI data and measured some goodness-of-fit statistics. The results are tabulated below:

Table 7.1: Results from four fitted distributions to the HDI data

Distributions	μ	σ	ν	GD	AIC
OLLLTN	0.6785535	0.1309002	0.2367922	-1167.249	-1161.249
SIMPLEX	0.6790461	0.7926546	-	-1104.26	-1100.26
LTN	0.6842778	0.3652149	-	-1098.859	-1094.859
BE	0.6790925	0.1663487	-	-1092.7	-1088.7

To estimate parameters for the data, we used the gamlss package to define an OLLLTN family from scratch. Through gamlss, we were able to get parameters for different distributions and their goodness of fit measures more easily than by using optim.

7.3 Remarks and Conclusions

- We fit four different distributions on the Brazilian HDI data, which is approximately bimodal in shape

- The *Global Deviance,* which is equal to two times the negative log likelihood, is the lowest for OLLLTN which indicates that the likelihood function is the highest for it among the other distributions

- The *Akaike Information Criterion (AIC)* is also a measure of the goodness of a fitting model. It takes into account the negative log likelihood as well as the number

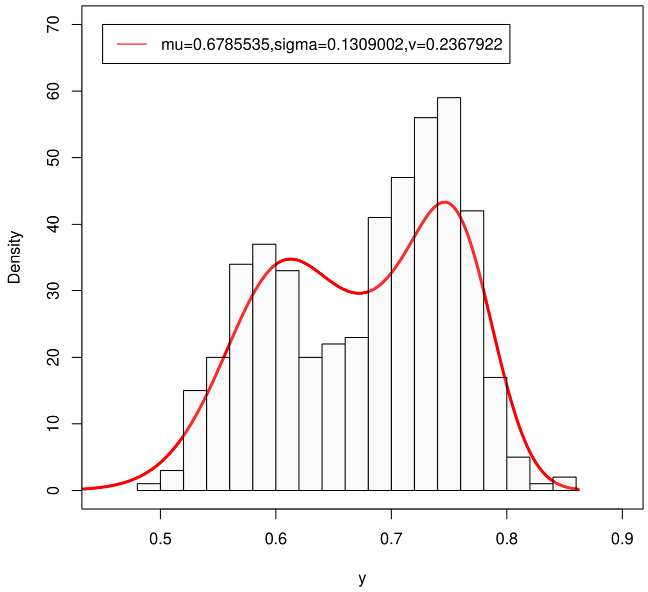

Figure 7.1: OLLLTN fitted to HDI data

of independent parameters used by the model. OLLLTN outperforms the others in this case as well

- The graphs also suggest that only OLLLTN distribution is able to account for the bimodality present in the data distribution. The other distributions are not able to account for the bimodality.

- The values in the table are in accordance to the values presented in the paper

Chapter 8

Discussion and Conclusions

- The author presented the OLLLTN distribution as an alternative for modelling data in the interval (0,1)
- First, we established the theoretical properties of the distribution such as pdf,cdf,moments,etc.
- We plotted the pdf of OLLLTN for various parameters and showed that it can be used to account for all kinds of bimodal data in the interval (0,1)
- Based on the Logit Normal Regression model, we defined the OLLLTN regression theoretically
- We conducted Monte Carlo Simulation studies to show that the parameters of a fitting-OLLLTN distribution can be correctly estimated for a given data
- We then simulated OLLLTN regression to show that we can correctly estimate the coeffecients of covariates in a regression setting for bimodal data
- Finally, we established the usefulness of the OLLLTN distribution by showing its application to real-world data, and that it even performs better than other distributions in some cases.
- We were successfully able to reproduce the trends of most of the simulation studies, graphs and tables in the paper
- The paper went a step furthur and performed OLLLTN regression on HDI data using educational and income measures as covariates

- However, we were unable to reproduce the regression on the HDI data as the authors had not posted the data of what they used as the covariates and we were unable to find the exact information anywhere on the internet
- We attempted to perform the regression using other similar data, but the results were not reproduced
- The main difficulty was that the regression was extremely unstable using optim, and often did not converge, especially when the coefficients were greater than one. However, on adding the link function, the convergence became stable and better results were obtained.

Chapter 9

Work Declaration

- Bipplav Kumar Tiwari - Finding and selecting the paper (60%), preparing the presentation and report structure and organising it (40%), handling of errors for small values of ν (30%), implementation of $OLLLTN$ family in GAMLSS (20%) implementation and testing of the results obtained by the various parts of the code (30%).
- Siddhant Shivdutt Singh - Suggesting the various important optimizations for a functional code (60%), finding the dataset from online sources and preprocessing it (50%), preparing the final report (10%), plotting various plots such as Kurtosis and skewness plots (30%).
- Tanush Kumar - Implementing the code in R language and testing/debugging it (80%), finding and selecting the paper (20%), preparing the presentation and report (20%), Suggesting the various important optimizations for a functional code (40%).
- Vineet Kumar - Finding and selecting the paper(20%), plotting and writitng codes for various functions such as pdf, cdf, sampling histograms plots (70%), preparing the presentation and report (30%), finding the dataset from online sources and preprocessing it (50%). Formatting of codes, presentation, and report.

  We highly enjoyed doing this project and would like to thank our supervisor, Dr. Arnab Hazra, for his constant support and guidance.

Appendix A

Python Script to make 3-D Plots

A.1 Kurtosis and Skewness

```
from mpl_toolkits import mplot3d
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
# Download data set from plotly repo
z_kurt_nu = pd.read_csv("z_kurt_nu.csv")
z_kurt_mu = pd.read_csv("z_kurt_mu.csv")
z_kurt_sigma = pd.read_csv("z_kurt_sigma.csv")

z_skew_nu = pd.read_csv("z_skew_nu.csv")
z_skew_mu = pd.read_csv("z_skew_mu.csv")
z_skew_sigma = pd.read_csv("z_skew_sigma.csv")

z_kurt_nu.dropna(inplace=True)
z_kurt_mu.dropna(inplace=True)
z_kurt_sigma.dropna(inplace=True)

z_skew_nu.dropna(inplace=True)
z_skew_mu.dropna(inplace=True)
z_skew_sigma.dropna(inplace=True)

z_skew_nu = z_skew_nu.drop(['Unnamed: 0'], axis=1)
z_skew_mu = z_skew_mu.drop(['Unnamed: 0'], axis=1)
z_skew_sigma = z_skew_sigma.drop(['Unnamed: 0'], axis=1)

z_kurt_nu = z_kurt_nu.drop(['Unnamed: 0'], axis=1)
z_kurt_mu = z_kurt_mu.drop(['Unnamed: 0'], axis=1)
z_kurt_sigma = z_kurt_sigma.drop(['Unnamed: 0'], axis=1)

n = 1000
x = np.outer(np.linspace(0.1, 1, n), np.ones(n))
```

```
# print(x)
y = x.copy().T
z_kurt_nu_val = z_kurt_nu.values[0:n,0:n]
z_kurt_mu_val = z_kurt_mu.values[0:n,0:n]
z_kurt_sigma_val = z_kurt_sigma.values[0:n,0:n]

z_skew_nu_val = z_skew_nu.values[0:n,0:n]
z_skew_mu_val = z_skew_mu.values[0:n,0:n]
z_skew_sigma_val = z_skew_sigma.values[0:n,0:n]

# print(z_nu_val.shape,x.shape,y.shape)
fig = plt.figure()
ax = plt.axes(projection ='3d')
ax.plot_surface(x,y,z_kurt_sigma_val, cmap ='viridis', edgecolor ='red')
ax.set_title('Surface plot for kurtosis : sigma constant')
ax.set(xlabel='mu axis', ylabel='nu axis', zlabel='kurt_sigma_val axis')
plt.show()
```

Bibliography

[1] J. C. S. Vasconcelos, F. Prataviera, E. M. M. Ortega, and G. M. Cordeiro, "An extended logit-normal regression with application to human development index data," *Communications in Statistics - Simulation and Computation*, 2022.

[2] C. Parada, *A collection of 79 attributes from Brazilian Cities*. 2022.

[3] M. Stasinopoulos, B. Rigby, and C. Akantziliotou, *Instructions on how to use the gamlss package in R*. 2008.

[4] M. Stasinopoulos, B. Rigby, V. Voudouris, C. Akantziliotou, M. Enea, and D. K. and, *Package 'gamlss'*. 2022.