An Introduction to Bayesian Quantile Regression for Binary Longutudinal Data with R Package `qbld`

Ayush Agarwal

August 29, 2020

Contents

1 Introduction

The R package `qbld` follows **Rahman and Vossmeyer (2019)** as its motivating literature, and contributes by extending the various methodologies in quantile framework, to a hierarchical Bayesian quantile regression model for binary longitudinal data (QBLD) and proposing a Markov chain Monte Carlo (MCMC) algorithm to estimate the model. The model handles both common (fixed) and individual-specific (random) parameters (commonly referred to as mixed effects in statistics). The algorithm implements a blocking, and an unblocking procedure that is computationally efficient and the distributions involved allow for easy calculations of covariate effects.

2 Quantile Regression for Binary Longitudinal Data

2.1 The Model

Let y be the reponse variable, and z be the introduced latent variable as described. The **QBLD** model can be conveniently expressed in the latent variable formulation (Albert & Chib, 1993) as follows:

$$\begin{aligned} z_{it} &= x_{it}^{'}\beta + s_{it}^{'}\alpha_i + \epsilon_{it}, \qquad \forall i = 1, ..., n; t = 1, ..., T_i \\ y_{it} &= \begin{cases} 1 & if\ z_{it} > 0 \\ 0 & \text{otherwise,} \end{cases} \\ \epsilon_{it} &= w_{it}\theta + \tau\sqrt{w_{it}}u_{it} \qquad \forall i = 1, ..., n; t = 1, ..., T_i \end{aligned} \tag{1}$$

y_{it} = response variable y at t^{th} time period for the i^{th} case,
z_{it} = unobserved latent variable z at t^{th} time period for the i^{th} case,
$x_{it} = k * 1$ vector of fixed-effects covariates,
$\beta = k * 1$ vector of fixed-effects parameters,
$s_{it} = l * 1$ vector of covariates that have case-specific effects,
$\alpha_i = l * 1$ vector of case-specific parameters, and
ϵ_{it} = the error term $\overset{\text{iid}}{\sim} AL(0, 1, p)$.

AL refers to the Asymmteric Laplace Distribution with location, $\mu = 0$, scale $\sigma = 1$, and skew paramter p. The error term is decomposed into a normal-exponential mixture representation of the AL distribution, presented in Kozumi and Kobayashi (2011).

Here, $u_{it} \sim N(0, 1)$, is mutually independent of $w_{it} \sim \exp(1)$, where $\exp(.)$ is the exponential distribution. Define: $\theta = \frac{1-2p}{p(1-p)}$, and $\tau = \sqrt{\frac{2}{p(1-p)}}$.

Random samples from the AL distribution are generated using `raldmix` function. (See Appendix)

2.2 Model priors

Longitudinal data models often involve a moderately large amount of data, so we stack the model for each case i.

We define, $z_i = (z_{i1}, ..., z_{iT_i})'$, $X_i = (x_{i1}, ..., x_{iT_i})$, $S_i = (s_{i1}, ..., s_{iT_i})$, $w_i = (w_{i1}, ..., w_{iT_i})'$, $D_{\tau\sqrt{w_i}} = diag(\tau\sqrt{w_{i1}}, ..., \tau\sqrt{w_{iT_i}})'$, and $u_i = (u_{i1}, ..., u_{iT_i})'$.

Building on Eq.(1),the resulting hierarchical model can be written as:

$$
\begin{aligned}
z_i &= X_i\beta + S_i\alpha_i + w_i\theta + D_{\tau\sqrt{w_i}}u_i \\
y_{it} &= \begin{cases} 1 & if\ z_{it} > 0 \\ 0 & \text{otherwise,} \end{cases} \\
\alpha_i|\varphi^2 &\sim N_l(0, \varphi^2 I_l), w_{it} \sim \exp(1), u_{it} \sim N(0,1) \\
\beta &\sim N_k(\beta_0, B_0), \varphi^2 \sim IG(c1/2, d1/2)
\end{aligned}
\tag{2}
$$

$IG(.)$ refers to the Inverse-Gamma distribution, exp(.) refers to the exponential distribution. The starting values for the sampler are sampled from the respective assumed priors, however, one is free to tweak β_0, B_0, $c1$, and $d1$ values.

2.3 Blocked vs Unblocked Sampler

The **unblocked** version of the Gibbs sampler is faster, but there is potential for poor mixing properties due to correlation between the covariates. I would recommend using "Unblock" for larger datasets. See Appendix for details of the algorithm.

To avoid potential slow mixing, an alternative **blocked** algorithm is presented. This algorithm however, takes a longer time to sample the Markov chain. I would recommend using "Block" for smaller datasets. See Appendix for details of the algorithm.

3 Using `qbld` package

Let us examine the dataset we will use to demonstrate the sample usage of the package.

3.1 Dataset:- Airpollution

This example datset is a subset of data from Six Cities study, a longitudinal study of the health effects of air pollution. The data set contains complete records on 537 children from Ohio, each child was examined annually at ages 7 through 10. The repeated binary response is the wheezing status (1="yes", 0="no") of a child at each occasion.

Each mother's smoking pattern was also recorded at the time of the study. Although mother's smoking status could vary with time, it was determined in the first interview and was treated as a time-independent covariate. Maternal smoking was categorized as 1 if the mother smoked regularly and 0 otherwise.

```
set.seed(10)
library(qbld)

## qbld: Quantile Regression for Binary Longitudinal Data
## Version 1.0 created on 2020-08-17.
##
## For citation information, type citation("qbld").
## Type help("qbld-package") or help("model.qbld") to get started.

data(airpollution)
str(airpollution)

## 'data.frame': 128 obs. of  5 variables:
##  $ id     : int  1 1 1 1 2 2 2 2 3 3 ...
##  $ wheeze : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ age    : num  7 8 9 10 7 8 9 10 7 8 ...
##  $ smoking: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ counts : num  237 237 237 237 10 10 10 10 15 15 ...
```

3.2 `model.qbld`: Running the QBLD sampler

`model.qbld` estimates the QBLD model as described in section (1), and outputs a `qbld` class object. In this example, we will model the wheezing status (`wheeze`) in terms of `age` and `smoking`. We will not treat `counts` as a covariate of interest, and allow intercepts for both fixed and random effects.

```
##modelling the output :- Blocked
#no burn, no verbose, no summary

output.block <- model.qbld(fixed_formula = wheeze~smoking+I(age^2),
                           data = airpollution, id="id",
                           random_formula = ~1, p=0.25,
                           nsim=1000, method="block", burn=0,
                           summarize=FALSE, verbose=FALSE)
```

Let us look at the arguments one by one:

- `fixed_formula:` A description of the model to be fitted of the form $response \sim fixed$ effects predictors i.e X_i in the model (2). Response variable is mandatory, and empty formula will throw error.

  In this example, $wheeze \sim smoking + I(age^2) + age$ translates to response variable, $y_i =$ `wheeze`, and x_i as `smoking, age,` age^2, *and* `Intercept`.

- `id:` An identifier variable in the dataset that specifies individual profile. Every row needs to contain an id value that maps the data point to the individual. By default, id = "id", and hence, data is expected to contain an id variable. Note that this is not a covariate, and is omitted while modelling.

- `data:` Data are contained in a `data.frame`. Each element of the data argument must be identifiable by a name. All subjects need to be observed at the same number of time points. Using datasets with different time points should be avoided. NAs are not allowed and should throw errors. All factor variables are auto-converted to numeric levels. Two datasets, `airpollution` and `locust` are built into the package.

- `random_formula:` A description of the model to be fitted of the form $response \sim random$ effects predictors i.e S_i in the model. Response

variable is not required, and is ignored. This defaults to S_i being only an intercept.

In this example, ~ 1 translates to s_i as `Intercept`.

- `p:` Quantile for the AL distribution on the error term, $p = 0.25$ by default. For very low (≤ 0.025) or very high (≥ 0.970) values of p, sampler forces to unblock version to avoid errors in the block procedure.

- `nsim:` No. of simulations to run the sampler.

- `b0, B0:` Prior model parameters for Beta as in the model (2). These are defaulted to 0 vector, and Identity matrix of appropriate dimensions. Full Gibbs Sampler is not affected by starting values, and need not be specified.

- `c1, d1:` Prior model parameters for Varphi2 as in the model (2). These are defaulted to 9, 10 (arbitrary) respectively. Full Gibbs Sampler is not affected by starting values, and need not be specified.

- `method:` Choose between the"Block" vs "Unblock" sampler, Block is slower, but produces lower correlation. Check section 3 for a detailed comparsion. I would recommend using "Unblock" for larger datasets. The code uses regex and is impervious to alphabet case related errors.

- `burn:` Burn in percentage, number between (0,1). Burn-in values are discarded while outputting and are not used for summary statistical calculations. No. of simulations are adjusted for burn-in before ESS calculations.

- `summarize:` False by default. Outputs a summary table (same as `summary(output)`). In addition to this, also prints Model fit diagonstics such as AIC, BIC, and Log-likelihood values. This is a bit unusual for a Bayesian analysis; however, useful to check alignment with the classical models or choose among quantile p values.

- `verbose:` False by default. If True, spits out progress reports while the sampler is running. This will print simulation progress for 10 times. i.e prints every 100th simulation if `nsim` = 1000.

3.3 `qbld` class object

The output of `model.qbld` function is a `qbld` class object.

```
str(output.block)

## List of 3
##  $ Beta   : num [1:1000, 1:3] 0 -0.122 -0.816 -1.816 -1.176 ...
##  $ Alpha  : num [1, 1:32, 1:1000] 0.0754 -1.9479 -0.6658 1.7591 -0.1183 ...
##  $ Varphi2: num [1:1000, 1] 1 0.627 0.86 1.015 1.693 ...
##  - attr(*, "burn")= logi FALSE
##  - attr(*, "nsim")= num 1000
##  - attr(*, "which")= chr "block"
##  - attr(*, "varnames")= chr [1:4] "(Intercept)" "smoking" "I(age^2)" "Varphi2"
##  - attr(*, "class")= chr "qbld"
##  - attr(*, "quantile")= num 0.25
```

`qbld` class object contains the following attributes:

- `Beta:` Matrix of MCMC samples of fixed-effects parameters.
- `Alpha:` 3-dimensional Matrix (of the form $\mathbb{R}^{k*l*m}$) of MCMC samples of random-effects parameters.
- `Varphi2:` Matrix of MCMC samples for varphi2.
- `nsim:` numeric; No. of simulations of MCMC.
- `burn:` logical; Whether or not burn-in used.
- `which:` Attribute; "block" or "unblock" sampler used

3.4 `summary.qbld`: Summarizing the `qbld` output

One way of summarizing the model is to use the `summarize` argument. Continuing with the example in the previous subsection, let us have a look at the unblocked sampler and understand the output.

```r
##modelling the output :- Unblocked
#Using burn, no verbose, and summary
# p = 0.50 i.e 50th quantile
output.unblock <- model.qbld(fixed_formula = wheeze~smoking+I(age^2)+age,
                             data = airpollution, id="id",
                             random_formula = ~1, p=0.50,
                             nsim=5000, method="Unblock", burn=0.2,
                             summarize=TRUE, verbose=FALSE)
```

```
## Please wait while we're processing your request.
## I recommend listening to Vienna by Billy Joel while you wait.
## https://music.apple.com/in/album/vienna/158617952?i=158618071
##
## Quantile used = 0.5
##
## No. of Iterations = 4000 samples
## Type of Sampler = unblock
## Burn-in Used? = TRUE
##
## 1. Statistics for each variable,
##                Mean   SD  MCSE  ESS Gelman-Rubin
## (Intercept)  0.017 0.98 0.016 4001     1.000000 *
## smoking     -0.036 0.51 0.023  503     1.000869
## I(age^2)     0.001 0.03 0.001  516     1.000844
## age         -0.003 0.36 0.014  665     1.000627
## Varphi2      1.063 0.45 0.021  477     1.000923
##
## MultiESS value = 682.1899
## Multi Gelman-Rubin = 1.000608
## Note : * indicates enough samples for the covariate
##        *** indicates enough samples for the whole sampler.
##
## 2. Quantiles for each variable,
##                2.5%    25%    50%   75% 97.5%
## (Intercept) -1.865 -0.675  0.033 0.666 1.930
## smoking     -1.037 -0.389 -0.033 0.308 0.985
## I(age^2)    -0.059 -0.020  0.001 0.022 0.061
## age         -0.705 -0.254 -0.005 0.240 0.714
## Varphi2      0.478  0.759  0.965 1.257 2.208
##
```

```
##
## 3. Model Selection Criterion
## Log likelihood = -77.49967
## AIC = 164.9993
## BIC = 181.3389
```

Note: that we are missing significance stars on the `Multi Gelman-Rubin` level as described in the output above. This is indicative of a lack of enough samples for MCMC. We will increase `nsim` to 20000 for the next run and try to achieve the significance level.

Let us also explore the second way of summarizing a `qbld` object through `summmary` S3 method, which produces a `qbld.summmary` class object.

```
##modelling the output :- Unblocked
#Using burn, no verbose, and summary

output.unblock2 <- model.qbld(fixed_formula = wheeze~smoking+I(age^2)+age,
                              data = airpollution, id="id",
                              random_formula = ~1, p=0.50,
                              nsim=20000, method="Unblock", burn=0.2,
                              summarize=FALSE, verbose=FALSE)
```

```
summary.unblock2 = summary(output.unblock2,
                           quantiles = c(0.025, 0.25, 0.5, 0.75, 0.975),
                           epsilon=0.10)
str(summary.unblock2)

## List of 9
##  $ statistics :'data.frame': 5 obs. of  6 variables:
##   ..$ Mean        : num [1:5] -0.002 0.014 0.001 -0.007 1.064
##   ..$ SD          : num [1:5] 0.97 0.52 0.03 0.36 0.452
##   ..$ MCSE        : num [1:5] 0.008 0.012 0.001 0.007 0.01
##   ..$ ESS         : num [1:5] 16000 2022 2176 2892 2049
##   ..$ Gelman-Rubin: num [1:5] 1 1 1 1 1
##   ..$             : chr [1:5] "*" "*" "*" "*" ...
##  $ quantiles  : num [1:5, 1:5] -1.914 -1.011 -0.058 -0.718 0.48 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:5] "(Intercept)" "smoking" "I(age^2)" "age" ...
```

```
##   .. ..$ : chr [1:5] "2.5%" "25%" "50%" "75%" ...
##  $ nsim       : num 16000
##  $ burn       : logi TRUE
##  $ which      : chr "unblock"
##  $ p          : num 0.5
##  $ multiess   : num 3063
##  $ multigelman: num 1
##  $ foo        : logi TRUE
##  - attr(*, "class")= chr "summary.qbld"
```

Note that the `foo` attribute is now TRUE, which means the significance on the `Multi Gelman-Rubin` level have been reached. Note that, in such a case, the summary table for this run will contain the stars unlike the last run.

`summary` function has the following arguments:

- `quantiles:` Vector of quantiles for summary of the covariates, defaulted to c(0.025, 0.25, 0.5, 0.75, 0.975)
- `epsilon:` 0.05 by default. Epsilon value is used for calculating `target.psrf` values, which estimate the ideal number of effective sample size required for a given level of significance. This value will be compared to generated ESS and significance stars are added accordingly. This process is repeated for individual chains and MultiESS, multi-Gelman by treating all the parameter chains as one multi-variate chain.

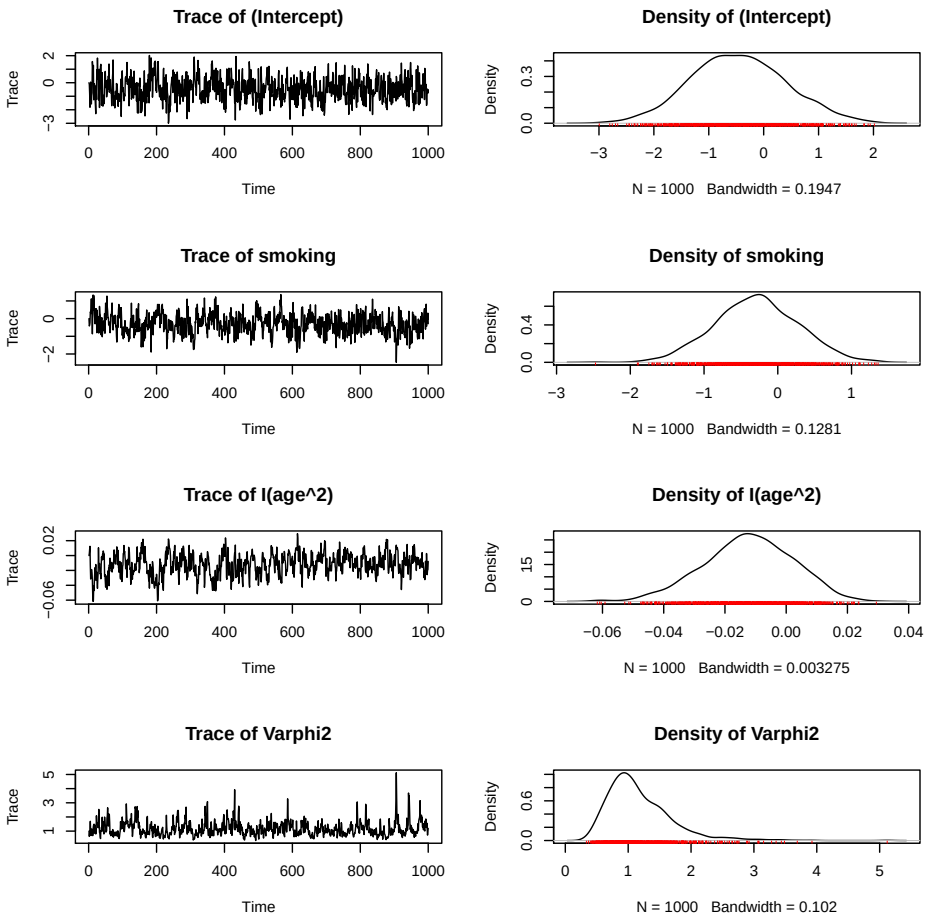
`qbld.summary` class object contains the following attributes:

- `statistics:` Contains the mean, sd, markov std error, ess and Gelman-Rubin diagnostic
- `quantiles:` Contains quantile estimates for each variable
- `nsim:` No. of simulations run, adjusted for burn-in
- `burn:` Burn-in used or not
- `which:` Block, or Unblock version of sampler
- `p:` quantile for the AL distribution on the error term
- `multiess:` multiess value for the sample
- `multigelman:` multivariate version of Gelman-Rubin

3.5 `plot.qbld`: Creating plots

Let us now try and create some diagnostic plots to understand the density spread of the covariate, as well as trace of the MCMC run.

```
par(mfrow=c(4,2))
plot(output.block, trace = TRUE, density = TRUE,
     auto.layout = FALSE, ask = NULL)
```



Plot function has the following arguments:

- `trace`: Whether or not to plot trace plots for covariates, TRUE by default

- `density:` Whether or not to plot density for covariates, TRUE by default.
- `auto.layout:` Auto set layout or not, TRUE as default. Plots according to the local settings if false.

4 Appendix

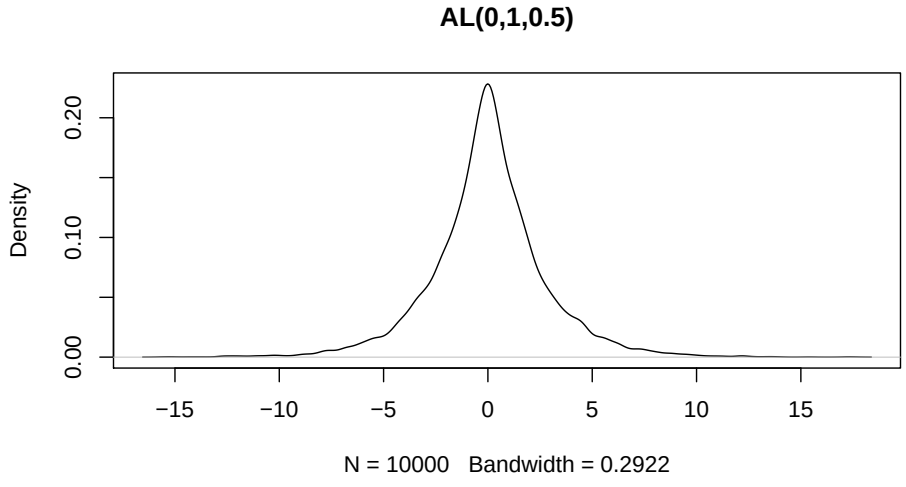
4.1 Asymmetric Laplace Distribution

The error term as described in (1) is a random variable from the AL distribution. For the sake of completeness, random generation and a few other AL functions have been made available to the user. For help using the functions, use `?aldmix`.
The asymmetric Laplace distribution (ALD), has the following pdf:

$$f(x;\mu,\sigma,p) = \frac{p(1-p)}{\sigma}\exp\{-\frac{(x-\mu)}{\sigma}(p - I(x \leq \mu))\} \tag{3}$$

where μ is the location paramter, σ is the scale parameter, and p is the skew paramter.

```
#generate 1e4 samples
ald.sample <- raldmix(n = 1e4, mu = 0, sigma = 1, p = 0.5)
plot(density(ald.sample), main="AL(0,1,0.5)")
```



```
## additional functions
ald.density <- daldmix(c(4,5),mu = 0,sigma = 1,p = 0.5)
ald.cdf <- paldmix(c(1,4),mu = 0,sigma = 1,p = 0.5,lower.tail=TRUE)
ald.quantile <- qaldmix(0.5,mu = 0,sigma = 1,p = 0.5,lower.tail=TRUE)
```

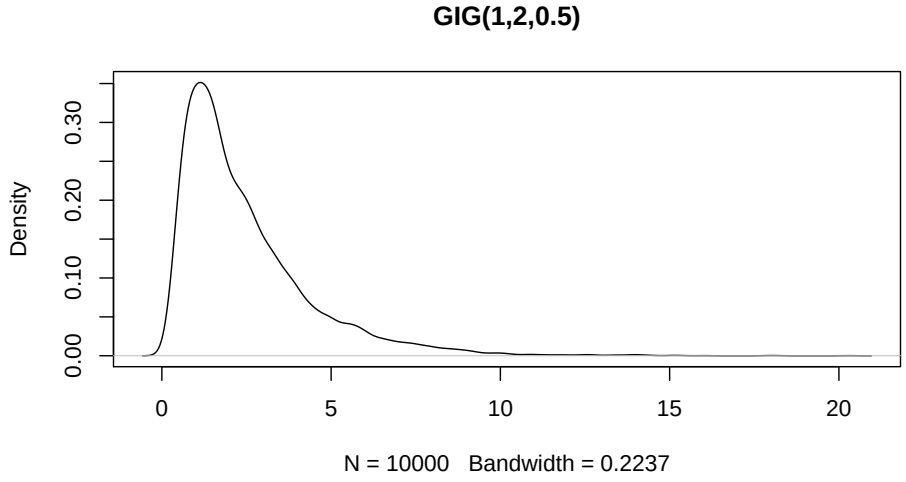
4.2 Generalized Inverse Gaussian Distribution

Gibbs sampler used in the model requires random sampling from Generalized Inverse Gaussian(GIG) distribution. For the sake of completeness, the random generation function `rgig`, and the density function, `dgig` are made available to the user. For help using the functions, use `?gig`.
The Generalised Inverse Gaussian distrubtion(GIG), which has the following pdf:

$$f(a,b,p) = \frac{(a/b)^{p/2}}{2K_p(\sqrt{ab})} \exp\{-\frac{ax+b/x}{2}\}, \quad x > 0 \tag{4}$$

where $a, b > 0$ and $p \in \mathbb{R}$ are the parameters, and K_p is a modified Bessel function of the second kind.

```
  # random generation
  gig.sample <- rgig(n = 1e4, lambda = 0.5, a = 1, b = 2)
  plot(density(gig.sample),main="GIG(1,2,0.5)")
```



```r
# density
gig.density <- dgig(x = 1, a = 1, b = 2, p = 0.5, log_density = FALSE)
```

4.3 Blocked Sampling

- Sample (β, z_i) in one block. These are sampled in following two sub-steps.
  - Sample β

$$
\begin{aligned}
\beta|z, w, \varphi^2 &\sim N(\tilde{\beta}, \tilde{B}), \\
where, \quad \tilde{B}^{-1} &= (\sum_{i=1}^{n} X_i' \Omega_i^{-1} X_i + B_0^{-1}), \\
\tilde{\beta} &= \tilde{B}(\sum_{i=1}^{n} X_i' \Omega_i^{-1}(z_i - w_i\theta) + B_0^{-1}\beta_0), \\
\Omega_i &= (\varphi^2 S_i S_i' + D^2_{\tau\sqrt{w_i}}).
\end{aligned}
\tag{5}
$$

  - Sample the vector $z_i|y_i, \beta, w_i, \varphi^2 \sim TMVN_{B_i}(X_i\beta + w_i\theta, \Omega_i)$ for all $i = 1, ..., n$, where $B_i = (B_{i1} * B_{i2} * ... * B_{iT_i})$ and B_{it} are interval $(0, \infty)$ if $y_{it} = 1$, and the interval $(-\infty, 0]$ if $y_{it} = 0$. This is done by sampling z_i at the j^{th} pass of the MCMC iteration using a series of conditional posteriors:

$$
\begin{aligned}
z_{it}^j|z_{i1}^j, ...z_{i(t-1)}^j, z_{i(t+1)}^{j-1}, ..., z_{iT_i}^{j-1} &\sim TN_{B_i}(\mu_{t|-t}, \Sigma_{t|-t}), \qquad t = 1, ..., T_i. \\
where, \quad \mu_{t|-t} &= x_{it}'\beta + w_{it}\theta + \Sigma_{t,-t}\Sigma_{-t,-t}^{-1}(z_{i,-t}^j - (X_i\beta + w_i\theta)_{-t}), \\
\Sigma_{t|-t} &= \Sigma_{t,t} - \Sigma_{t,-t}\Sigma_{-t,-t}^{-1}\Sigma_{-t,t},
\end{aligned}
\tag{6}
$$

    where $z_{i,-t}^j = (z_{i1}^j, ...z_{i(t-1)}^j, z_{i(t+1)}^{j-1}, ..., z_{iT_i}^{j-1})$, $(X_i\beta + w_i\theta)_{-t}$ is column vector with t^{th} element removed, $\Sigma_{t,t}, \Sigma_{t,-t}, \Sigma_{-t,-t}$ are $(t,t)^{th}$ element, t^{th} row with t^{th} element removed, and t^{th} row and column removed respectively.

- Sample α

$$\begin{aligned} \alpha_i|z,\beta,w,\varphi^2 &\sim N(\tilde{a},\tilde{A}), \quad \forall i=1,...,n \\ where, \quad \tilde{A^{-1}} &= (S_i^{'}D^{-2}_{\tau\sqrt{w_i}}S_i+\frac{1}{\varphi^2}I_l), \\ \tilde{a} &= \tilde{A}(S_i^{'}D^{-2}_{\tau\sqrt{w_i}}(z_i-X_i\beta-w_i\theta)). \end{aligned} \tag{7}$$

- Sample w

$$\begin{aligned} w_{it}|z_{it},\beta,\alpha_i &\sim GIG(0.5,\tilde{\lambda_{it}},\tilde{\eta}) \quad \forall i=1,...,n; t=1,...,T_i, \\ where, \quad \tilde{\lambda_{it}} &= (\frac{z_{it}-x_{it}^{'}\beta-s_{it}^{'}\alpha_i}{\tau})^2 \\ \tilde{\eta} &= (\frac{\theta^2}{\tau^2}+2). \end{aligned} \tag{8}$$

- Sample φ^2

$$\begin{aligned} \varphi^2|\alpha &\sim IG(\tilde{c_1}/2,\tilde{d_1}/2), \\ where, \quad \tilde{c_1} &= (nl+c_1), \\ \tilde{d_1} &= (\sum_{i=1}^{n}\alpha_i^{'}\alpha_i+d_1). \end{aligned} \tag{9}$$

.

4.4 Unblocked Sampling

- Sample β

$$\begin{aligned} \beta|z,w,\varphi^2 &\sim N(\tilde{\beta},\tilde{B}), \\ where, \quad \tilde{B}^{-1} &= (\sum_{i=1}^{n}X_i^{'}\Psi_i^{-1}X_i+B_0^{-1}), \\ \tilde{\beta} &= \tilde{B}(\sum_{i=1}^{n}X_i^{'}\Psi_i^{-1}(z_i-w_i\theta-S_i\alpha_i)+B_0^{-1}\beta_0), \\ \Psi_i &= D^2_{\tau\sqrt{w_i}}. \end{aligned} \tag{10}$$

- Sample α as in (7).
- Sample w as in (8).
- Sample φ^2 as in (9).
- Sample $z|y, \alpha, w \ \forall i = 1, ..., n; t = 1, ..., T_i$, from univariate truncated normal as:

$$z_{it}|y, \beta, w = \begin{cases} TN_{(-\infty,0]}(x_{it}^{'}\beta + s_{it}^{'}\alpha_i + w_{it}\theta, \tau^2 w_{it}) & if\, y_{it} = 0 \\ TN_{(0,\infty)}(x_{it}^{'}\beta + s_{it}^{'}\alpha_i + w_{it}\theta, \tau^2 w_{it}) & if\, y_{it} = 1 \end{cases} \tag{11}$$

5 References

- Rahman, Mohammad & Vossmeyer, Angela. (2018). Estimation and Applications of Quantile Regression for Binary Longitudinal Data. Advances in Econometrics. 40.
- Vats, Dootika and Christina Knudson. "Revisiting the Gelman-Rubin Diagnostic." arXiv: Computation (2018): n. pag.
- Keming Yu & Jin Zhang (2005) A Three-Parameter Asymmetric Laplace Distribution and Its Extension, Communications in Statistics - Theory and Methods.
- Kobayashi, Genya. (2011). Gibbs Sampling Methods for Bayesian Quantile Regression. J Stat Comput Simul.
- Devroye, L. Random variate generation for the generalized inverse Gaussian distribution. Stat Comput 24, 239–246 (2014).
- Wolfgang Hörmann and Josef Leydold (2013). Generating generalized inverse Gaussian random variates, Statistics and Computing.
- J. S. Dagpunar (1989). An easily implemented generalised inverse Gaussian generator, Comm. Statist. B – Simulation Comput. 18, 703–710.