QBLD - GSoC 2020

Ayush Agarwal

24/07/2020

Quantile Regression and the Asymmetric Lalplace Distribution

The Model

The QBLD model can be conveniently expressed in the latent variable formulation (Albert & Chib, 1993) as follows:

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

y_{it} = response variable y at t^{th} time period for the i^{th} individual,
z_{it} = unobserved latent variable z at t^{th} time period for the i^{th} individual,
$x_{it}^{'} = 1 \times k$ vector of fixed-effects covariates,
$\beta = k \times 1$ vector of fixed-effects parameters,
$s_{it}^{'} = 1 \times l$ vector of covariates that have individual-specific effects,
$\alpha_i = l \times 1$ vector of individual-specific parameters, and
ϵ_{it} = the error term $\overset{\text{iid}}{\sim} AL(0, 1, p)$.

ALD Mixture

While working directly with the AL density is an option, the resulting posterior will not yield the full set of tractable conditional distributions necessary for a Gibbs sampler. The mixture representation gives access to the appealing properties of the normal distribution. Thus, we utilize the normal-exponential mixture representation of the AL distribution, presented in Kozumi and Kobayashi (2011) :

$$\epsilon_{it} = w_{it}\theta + \tau\sqrt{w_{it}}u_{it} \qquad \forall i = 1, ..., n; t = 1, ..., T_i \tag{2}$$

$u_{it} \sim N(0, 1)$, is mutually inependent of $w_{it} \sim \exp(1)$,
$\theta = \frac{1-2p}{p(1-p)}$, and $\tau = \sqrt{\frac{2}{p(1-p)}}$.

Model with Priors

Longitudinal data models often involve a moderately large amount of data, so it is important to take advantage of any opportunity to reduce the computational burden. One such trick is to stack the model for each individual i (Hendricks, Koenker, & Poirier, 1979).

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 Eqs. (1) and (2),

$$\begin{gathered} z_i = X_i\beta + S_i\alpha_i + w_i\theta + D_{\tau\sqrt{w_i}}u_i, \\ y_{it} = \begin{cases} 1 & if\, z_{it} > 0 \\ 0 & \text{otherwise}, \end{cases} \\ \alpha_i|\varphi^2 \sim N_l(0, \varphi^2 I_l), w_{it} \sim \exp(1), u_{it} \sim N(0, 1) \\ \beta \sim N_k(\beta_0, B_0), \varphi^2 \sim IG(c1/2, d1/2) \end{gathered} \tag{3}$$

Algorithm

Blocked Sampling

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

  (1) Sample β

$$\begin{gathered} \beta|z, w, \varphi^2 \sim N(\tilde{\beta}, \tilde{B}), \\ where, \quad \tilde{B}^{-1} = (\sum_{i=1}^{n} X'_i\Omega_i^{-1}X_i + B_0^{-1}), \\ \tilde{\beta} = \tilde{B}(\sum_{i=1}^{n} X'_i\Omega_i^{-1}(z_i - w_i\theta) + B_0^{-1}\beta_0), \\ \Omega_i = (\varphi^2 S_i S'_i + D^2_{\tau\sqrt{w_i}}). \end{gathered} \tag{4}$$

(2) 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:

$$\begin{aligned} z_{it}^j | z_{i1}^j, ...z_{i(t-1)}^j, z_{i(t+1)}^{j-1}, ..., z_{iT_i}^{j-1} &\sim TN_{B_i}(\mu_{t|-t}, \Sigma_{t|-t}), \qquad t = 1, ..., T_i. \\ where, \quad \mu_{t|-t} &= x_{it}^{'}\beta + w_{it}\theta + \Sigma_{t,-t}\Sigma_{-t,-t}^{-1}(z_{i,-t}^j - (X_i\beta + w_i\theta)_{-t}), \\ \Sigma_{t|-t} &= \Sigma_{t,t} - \Sigma_{t,-t}\Sigma_{-t,-t}^{-1}\Sigma_{-t,t}, \end{aligned} \tag{5}$$

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

$$\begin{aligned} \alpha_i | z, \beta, w, \varphi^2 &\sim N(\tilde{a}, \tilde{A}), \quad \forall i = 1, ..., n \\ where, \quad \tilde{A^{-1}} &= (S_i^{'} D_{\tau\sqrt{w_i}}^{-2} S_i + \frac{1}{\varphi^2} I_l), \\ \tilde{a} &= \tilde{A}(S_i^{'} D_{\tau\sqrt{w_i}}^{-2}(z_i - X_i\beta - w_i\theta)). \end{aligned} \tag{6}$$

- Sample w

$$\begin{aligned} w_{it} | z_{it}, \beta, \alpha_i &\sim GIG(0.5, \tilde{\lambda_{it}}, \tilde{\eta}) \quad \forall i = 1, ..., n; t = 1, ..., T_i, \\ where, \quad \tilde{\lambda_{it}} &= (\frac{z_{it} - x_{it}^{'}\beta - s_{it}^{'}\alpha_i}{\tau})^2 \\ \tilde{\eta} &= (\frac{\theta^2}{\tau^2} + 2). \end{aligned} \tag{7}$$

- Sample φ^2

$$\begin{aligned} \varphi^2 | \alpha &\sim IG(\tilde{c_1}/2, \tilde{d_1}/2), \\ where, \quad \tilde{c_1} &= (nl + c_1), \\ \tilde{d_1} &= (\sum_{i=1}^{n} \alpha_i^{'}\alpha_i + d_1). \end{aligned} \tag{8}$$

.

Unblocked Sampling

- Sample β

$$
\begin{aligned}
&\beta|z, w, \varphi^2 \sim N(\tilde{\beta}, \tilde{B}),\\
&where, \quad \tilde{B}^{-1} = (\sum_{i=1}^{n} X_i^{'} \Psi_i^{-1} X_i + B_0^{-1}),\\
&\tilde{\beta} = \tilde{B}(\sum_{i=1}^{n} X_i^{'} \Psi_i^{-1} (z_i - w_i\theta - S_i\alpha_i) + B_0^{-1}\beta_0),\\
&\Psi_i = D^2_{\tau\sqrt{w_i}}.
\end{aligned}
\tag{9}
$$

- Sample α as in (6).
- Sample w as in (7).
- Sample φ^2 as in (8).
- Sample $z|y, \alpha, w \ \forall i = 1, ..., n; t = 1, ..., T_i$, from univariate truncated normal as:

$$
z_{it}|y, \beta, w = \begin{cases} TN_{(-\infty,0]}(x_{it}^{'}\beta + s_{it}^{'}\alpha_i + w_{it}\theta, \tau^2 w_{it}) & if\, y_{it} = 0 \\ TN_{(0,\infty)}(x_{it}^{'}\beta + s_{it}^{'}\alpha_i + w_{it}\theta, \tau^2 w_{it}) & if\, y_{it} = 1 \end{cases}
\tag{10}
$$

.

qbild_update

Ayush Agarwal

12/08/2020

How to get Qbild?

- Download the qbldcpp folder from the **GitHub repo**,
- Run the following commands :-
  - R CMD build qbild
  - R CMD install qbildcpp_1.0.tar.gz

After finishing the steps:

```
library(qbild)
library(knitr)

set.seed(10)

##############
## Loading and manipulation of the data set
##############

data <- readRDS("~/airpollution.rda") #contains factor variables as well

nsim = 5000
p = 0.25

##with intercept
time_a = Sys.time()
out <- model.qbld(data, id = "id", fixed_formula = wheeze~age+I(age^2)+smoking+counts,
                  random_formula = ~counts , nsim, p = 0.25, summarize = TRUE,
                  verbose = FALSE)  #intercept true for both, added manipulation of symbols
```

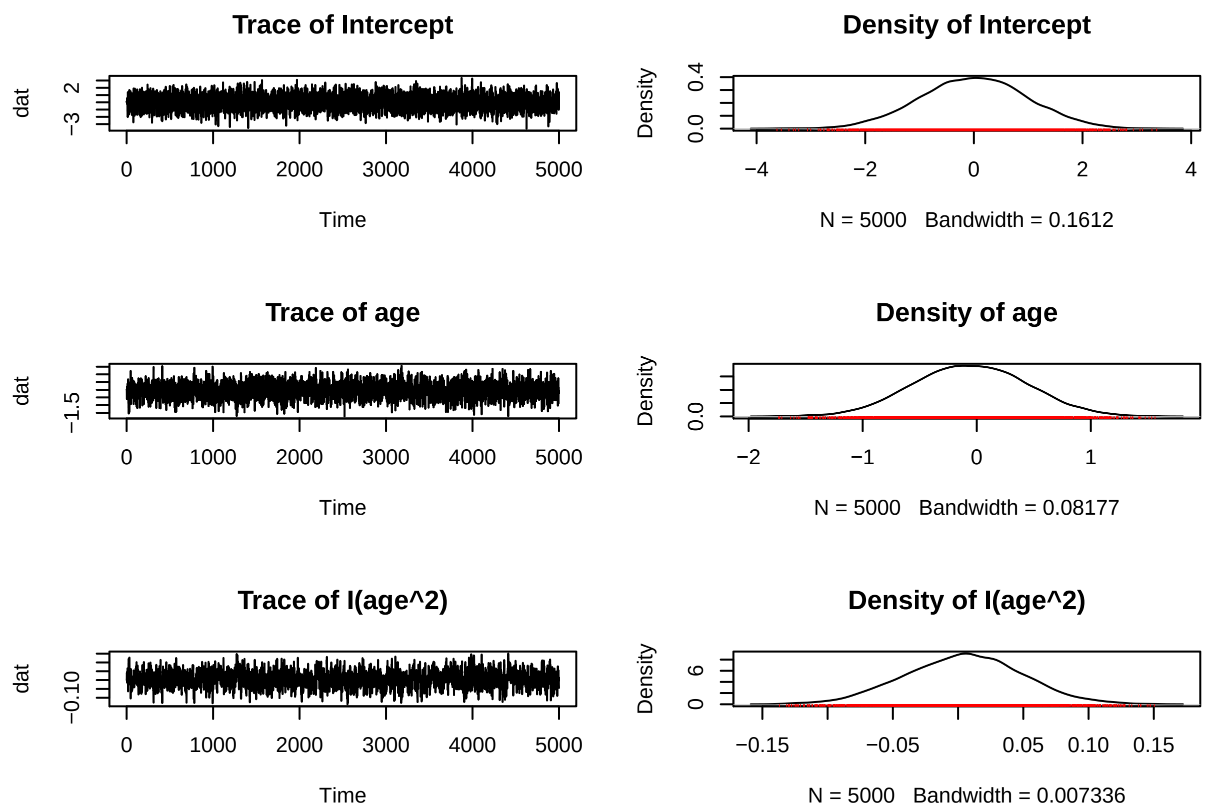
```
##
## Quantile used = 0.25
##
## No. of Iterations = 5000 samples
## Type of Sampler = block
## Burn-in Used? = FALSE
##
## 1. Statistics for each variable,
##             Mean   SD  MCSE     ESS GR Diagnostic
## Intercept  0.00 0.98 0.014 5108.43      1.000029
```

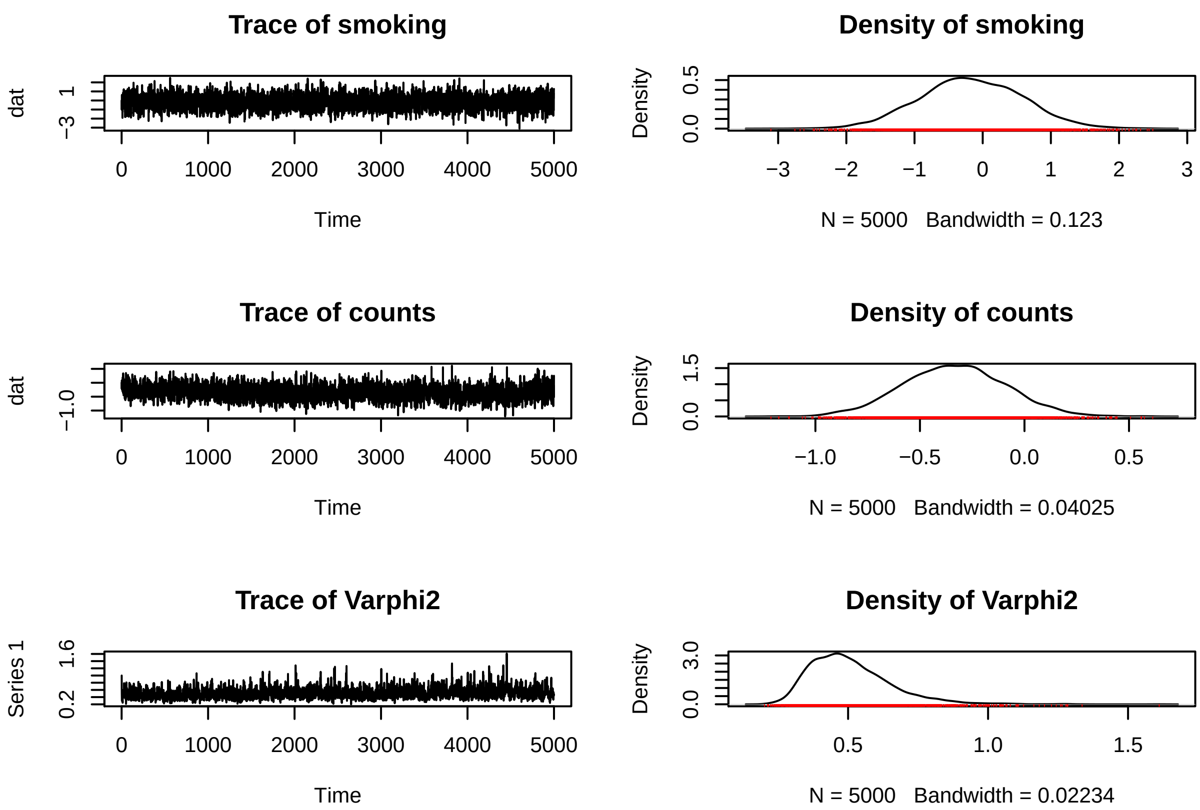
```
## age       -0.06 0.50 0.014 1272.38      1.000184
## I(age^2)   0.00 0.04 0.002  751.11      1.000359
## smoking   -0.17 0.75 0.011 4308.79      1.000040
## counts    -0.33 0.25 0.014  288.44      1.000751
## Varphi2    0.51 0.15 0.005  737.29      1.000615
##
##
## 2. Quantiles for each variable,
##             2.5%    25%    50%    75% 97.5%
## Intercept -1.930 -0.650  0.010  0.669 1.932
## age       -1.024 -0.397 -0.055  0.290 0.915
## I(age^2)  -0.083 -0.026  0.005  0.034 0.092
## smoking   -1.638 -0.671 -0.184  0.346 1.296
## counts    -0.814 -0.503 -0.337 -0.170 0.143
## Varphi2    0.302  0.402  0.485  0.585 0.854
##
## MultiESS value = 2043.821 737.2893
##
## 3. Model Selection Criterion
## Log likelihood = -71.46137
## AIC = 154.9227
## BIC = 177.5801
```

```
time_b = Sys.time()
paste0("Time elapsed = ",round(time_b-time_a,2)," sec")
```

```
## [1] "Time elapsed = 2.99 sec"
```

```
plot(out)
```


Trace of Intercept
dat
2
−3
0
1000
2000
3000
4000
5000
Time
Density of Intercept
Density
0.0
0.4
−4
−2
0
2
4
N = 5000   Bandwidth = 0.1612
Trace of age
dat
−1.5
0
1000
2000
3000
4000
5000
Time
Density of age
Density
0.0
−2
−1
0
1
N = 5000   Bandwidth = 0.08177
Trace of I(age^2)
dat
−0.10
0
1000
2000
3000
4000
5000
Time
Density of I(age^2)
Density
0
6
−0.15
−0.05
0.05
0.10
0.15
N = 5000   Bandwidth = 0.007336




```r
##with intercept
time_a = Sys.time()
out2 <- model.qbld(data, id = "id", fixed_formula = wheeze~age+smoking+counts-1,
              random_formula = ~. , nsim, p = 0.25, summarize = TRUE, method="Unblock",
                  verbose = FALSE)  #intercept true foronly random
```

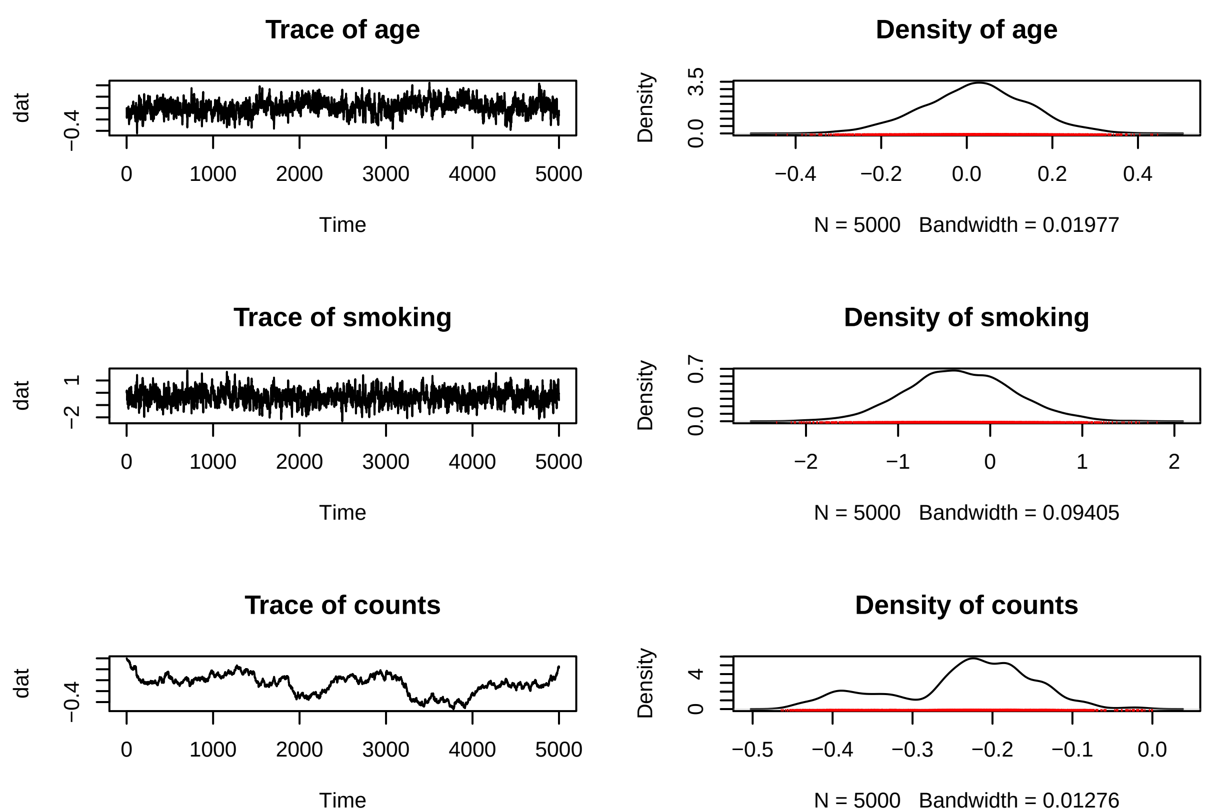
```
##
## Quantile used = 0.25
##
## No. of Iterations = 5000 samples
## Type of Sampler = Unblock
## Burn-in Used? = FALSE
##
## 1. Statistics for each variable,
##          Mean   SD  MCSE    ESS GR Diagnostic
## age      0.02 0.12 0.012 107.89      1.011434
## smoking -0.33 0.58 0.028 433.92      1.001291
## counts  -0.24 0.09 0.045   3.89      1.055949
## Varphi2  1.00 0.44 0.021 461.51      1.000969
##
##
## 2. Quantiles for each variable,
##           2.5%    25%    50%    75%  97.5%
## age     -0.229 -0.058  0.024  0.104  0.264
## smoking -1.436 -0.712 -0.341  0.057  0.842
## counts  -0.423 -0.280 -0.222 -0.176 -0.088
```

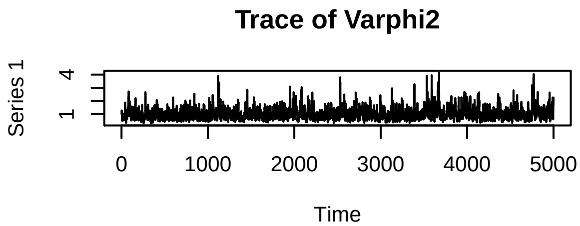
```
## Varphi2  0.448  0.696  0.915  1.186  2.120
##
## MultiESS value = 76.5387 461.5091
##
## 3. Model Selection Criterion
## Log likelihood = -77.05763
## AIC = 162.3068
## BIC = 175.6028
```

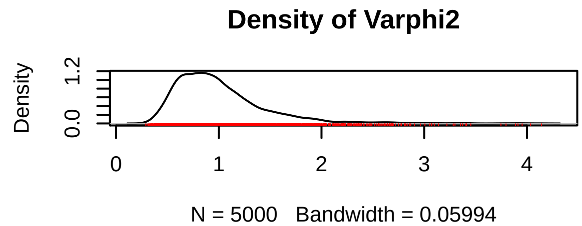
```
time_b = Sys.time()
paste0("Time elapsed = ",round(time_b-time_a,2)," sec")
```

```
## [1] "Time elapsed = 1.54 sec"
```

```
plot(out2)
```




Trace of Varphi2
Series 1
4
1
0
1000
2000
3000
4000
5000
Time


Density of Varphi2
Density
1.2
0.0
0
1
2
3
4
N = 5000   Bandwidth = 0.05994