lqmm: Estimating Quantile Regression Models for Independent and Hierarchical Data with R

Marco Geraci

MRC Centre of Epidemiology for Child Health
Institute of Child Health, University College London

m.geraci@ich.ucl.ac.uk

useR! 2011
August 16-18, 2011
University of Warwick, Coventry, UK

Quantile regression

Conditional quantile regression (QR) pertains to the estimation of unknown quantiles of an outcome as a function of a set of covariates and a vector of fixed regression coefficients.

For example, consider a sample of 654 observations of FEV1 in individuals aged 3 to 19 years who were seen in the Childhood Respiratory Disease (CRD) Study in East Boston, Massachusetts [1]. We might be interested in estimating **median** FEV1 or any other **quantile** as a **function** of age, sex, smoking, etc.

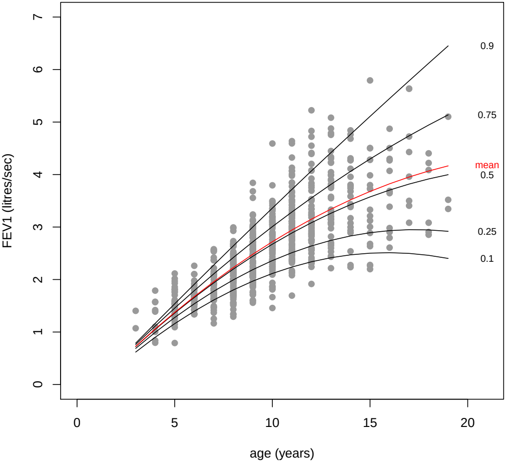
[1] Data available at http://www.statsci.org/datasets.html

Quantile regression

Conditional quantile regression (QR) pertains to the estimation of unknown quantiles of an outcome as a function of a set of covariates and a vector of fixed regression coefficients.

For example, consider a sample of 654 observations of FEV1 in individuals aged 3 to 19 years who were seen in the Childhood Respiratory Disease (CRD) Study in East Boston, Massachusetts [1]. We might be interested in estimating **median** FEV1 or any other **quantile** as a **function** of age, sex, smoking, etc.

[1] Data available at http://www.statsci.org/datasets.html



Regression quantiles (black) and mean fit (red) of FEV1 vs Age.

Quantile regression (contd)

Let's index the quantiles Q of the continuous response y_i with p, $0 < p < 1$, that is $\Pr(y_i \leqslant Q_{y_i}(p)) = p$. The conditional (linear) quantile function

$$Q_{y_i}(p|x_i) = x_i'\beta(p), \qquad i = 1, \ldots, N$$

can be estimated by solving (Koenker and Bassett, *Econometrica*, 1978)

$$\min_{\beta} \sum_i g_p(y_i - x_i'\beta(p)),$$

where $g_p(z) = z(p - I(z < 0))$ and $\beta(p)$ is the regression coefficient vector indexed by p.

Quantile regression (contd)

Let's index the quantiles Q of the continuous response y_i with p, $0 < p < 1$, that is $\Pr(y_i \leqslant Q_{y_i}(p)) = p$.The conditional (linear) quantile function

$$Q_{y_i}(p|x_i) = x_i'\beta(p), \qquad i = 1, \ldots, N$$

can be estimated by solving (Koenker and Bassett, *Econometrica*, 1978)

$$\min_{\beta} \sum_i g_p(y_i - x_i'\beta(p)),$$

where $g_p(z) = z(p - I(z < 0))$ and $\beta(p)$ is the regression coefficient vector indexed by p.

Quantile regression (contd)

Let's index the quantiles Q of the continuous response y_i with p, $0 < p < 1$, that is $\Pr(y_i \leqslant Q_{y_i}(p)) = p$.The conditional (linear) quantile function

$$Q_{y_i}(p|x_i) = x_i'\beta(p), \qquad i = 1, \ldots, N$$

can be estimated by solving (Koenker and Bassett, *Econometrica*, 1978)

$$\min_{\beta} \sum_i g_p\left(y_i - x_i'\beta(p)\right),$$

where $g_p(z) = z\,(p - I(z < 0))$ and $\beta(p)$ is the regression coefficient vector indexed by p.

Quantile regression (contd)

Let's index the quantiles Q of the continuous response y_i with p, $0 < p < 1$, that is $\Pr(y_i \leqslant Q_{y_i}(p)) = p$.The conditional (linear) quantile function

$$Q_{y_i}(p|x_i) = x_i'\beta(p), \qquad i = 1, \ldots, N$$

can be estimated by solving (Koenker and Bassett, *Econometrica*, 1978)

$$\min_{\beta} \sum_i g_p\left(y_i - x_i'\beta(p)\right),$$

where $g_p(z) = z\left(p - I(z < 0)\right)$ and $\beta(p)$ is the regression coefficient vector indexed by p.

Likelihood-based quantile regression: The asymmetric Laplace

Mean regression problem
(least squares)

$$\min_{\beta} \sum (y - x'\beta)^2$$

$\updownarrow$

$$\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(y - x'\beta)^2\right\}$$

Normal distribution

Quantile regression problem
(least absolute deviations)

$$\min_{\beta} \sum g_p(y - x'\beta)$$

$\updownarrow$

$$\frac{p(1-p)}{\sigma} \exp\left\{-\frac{1}{\sigma} g_p(y - x'\beta)\right\}$$

Asymmetric Laplace distribution

Likelihood-based quantile regression: The asymmetric Laplace

Mean regression problem
(least squares)

$$\min_{\beta} \sum (y - x'\beta)^2$$

Quantile regression problem
(least absolute deviations)

$$\min_{\beta} \sum g_p(y - x'\beta)$$

location parameter

$$\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{1}{2\sigma^2}(y - x'\beta)^2 \right\}$$

Normal distribution

$$\frac{p(1-p)}{\sigma} \exp\left\{ -\frac{1}{\sigma} g_p(y - x'\beta) \right\}$$

Asymmetric Laplace distribution

Likelihood-based quantile regression: The asymmetric Laplace

Mean regression problem
(least squares)

$$\min_{\beta} \sum (y - x'\beta)^2$$

Quantile regression problem
(least absolute deviations)

$$\min_{\beta} \sum g_p(y - x'\beta)$$

scale parameter

$$\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(y - x'\beta)^2\right\}$$

Normal distribution

$$\frac{p(1-p)}{\sigma} \exp\left\{-\frac{1}{\sigma} g_p(y - x'\beta)\right\}$$

Asymmetric Laplace distribution

Likelihood-based quantile regression: The asymmetric Laplace

Mean regression problem
(least squares)

$$\min_{\beta} \sum (y - x'\beta)^2$$

Quantile regression problem
(least absolute deviations)

$$\min_{\beta} \sum g_p(y - x'\beta)$$

skewness parameter

$$\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{1}{2\sigma^2}(y - x'\beta)^2 \right\}$$

Normal distribution

$$\frac{p(1-p)}{\sigma} \exp\left\{ -\frac{1}{\sigma} g_p(y - x'\beta) \right\}$$

Asymmetric Laplace distribution

Quantile regression and random effects

If $y \sim \mathrm{AL}(\mu, \sigma, p)$ then $Q_y(p) = \mu$.

The aim is to develop a QR model for hierarchical data. Inclusion of random intercepts in the conditional quantile function is straightforward (Geraci and Bottai, *Biostatistics*, 2007)

$$Q_y(p|x, u) = x'\beta(p) + u.$$

Likelihood-based estimation (MCEM – R and WinBUGS) assuming

- $y = X\beta + Zu + \epsilon$
- $u \sim \mathrm{N}\left(0, \sigma_u^2\right)$
- $\epsilon \sim \mathrm{AL}\left(0, \sigma I, p\right)$ (p is fixed a priori)
- $u \perp \epsilon$

Quantile regression and random effects

If $y \sim \mathrm{AL}(\mu, \sigma, p)$ then $Q_y(p) = \mu$.

The aim is to develop a QR model for hierarchical data. Inclusion of random intercepts in the conditional quantile function is straightforward (Geraci and Bottai, *Biostatistics*, 2007)

$$Q_y(p|x, u) = x'\beta(p) + u.$$

Likelihood-based estimation (MCEM – R and WinBUGS) assuming

- $y = X\beta + Zu + \epsilon$
- $u \sim \mathrm{N}(0, \sigma_u^2)$
- $\epsilon \sim \mathrm{AL}(0, \sigma I, p)$ (p is fixed a priori)
- $u \perp \epsilon$

Quantile regression and random effects

If $y \sim \mathrm{AL}(\mu, \sigma, p)$ then $Q_y(p) = \mu$.

The aim is to develop a QR model for hierarchical data. Inclusion of random intercepts in the conditional quantile function is straightforward (Geraci and Bottai, *Biostatistics*, 2007)

$$Q_y(p|x, u) = x'\beta(p) + u.$$

Likelihood-based estimation (MCEM – R and WinBUGS) assuming

- $y = X\beta + Zu + \epsilon$
- $u \sim \mathrm{N}\left(0, \sigma_u^2\right)$
- $\epsilon \sim \mathrm{AL}(0, \sigma I, p)$ (p is fixed a priori)
- $u \perp \epsilon$

Linear Quantile Mixed Models

The package lqmm (S3-style) is a suite of commands for fitting linear quantile mixed models of the type

- $y = X\beta + Zu + \epsilon$
- continuous y
- two-level nested model (e.g., repeated measurements on same subject, households within same postcode, etc)
- $\epsilon \sim \mathrm{AL}(0, \sigma I, p)$
- **multiple**, symmetric random effects with covariance matrix Ψ $(q \times q)$
- $u \perp \epsilon$

Note all unknown parameters are p-dependent

Linear Quantile Mixed Models

The package lqmm (S3-style) is a suite of commands for fitting linear quantile mixed models of the type

- $y = X\beta + Zu + \epsilon$
- continuous y
- two-level nested model (e.g., repeated measurements on same subject, households within same postcode, etc)
- $\epsilon \sim \mathrm{AL}\left(0, \sigma I, p\right)$
- **multiple**, symmetric random effects with covariance matrix Ψ $(q \times q)$
- $u \perp \epsilon$

Note all unknown parameters are p-dependent

LQMM estimation

Let the pair (ij), $j = 1, \ldots, n_i$, $i = 1, \ldots, M$, index the j-th observation for the i-th cluster/group/subject. The joint density of (y, u) based on M clusters for the linear quantile mixed model is given by

$$f(y, u|\beta, \sigma, \Psi) = f(y|\beta, \sigma, u)f(u|\Psi) = \prod_{i=1}^{M} f(y_i|\beta, \sigma, u_i)f(u_i|\Psi)$$

Numerical integration of likelihood (log-concave by Prékopa, 1973)

$$L_i(\beta, \sigma, \Psi|y) = \sigma_{n_i}(p) \int_{R^q} \exp\left\{-\frac{1}{\sigma} g_p\left(y_i - x_i'\beta(p) - z_i'u_i\right)\right\} f(u_i|\Psi)du_i,$$

where $\sigma_{n_i}(p) = [p(1-p)/\sigma]^{n_i}$ and $g_p(e_i) = \sum_{j=1}^{n_i} g_p(e_j)$.

LQMM estimation

Let the pair (ij), $j = 1, \ldots, n_i$, $i = 1, \ldots, M$, index the j-th observation for the i-th cluster/group/subject. The joint density of (y, u) based on M clusters for the linear quantile mixed model is given by

$$f(y, u|\beta, \sigma, \Psi) = f(y|\beta, \sigma, u)f(u|\Psi) = \prod_{i=1}^{M} f(y_i|\beta, \sigma, u_i)f(u_i|\Psi)$$

Numerical integration of likelihood (log-concave by Prékopa, 1973)

$$L_i(\beta, \sigma, \Psi|y) = \sigma_{n_i}(p) \int_{R^q} \exp\left\{-\frac{1}{\sigma} g_p\left(y_i - x_i'\beta\left(p\right) - z_i' u_i\right)\right\} f(u_i|\Psi)du_i,$$

where $\sigma_{n_i}(p) = [p(1-p)/\sigma]^{n_i}$ and $g_p\left(e_i\right) = \sum_{j=1}^{n_i} g_p\left(e_j\right)$.

Numerical integration with

- normal random effects $u \sim \mathrm{N}(0, \Psi)$ $\rightarrow$ Gauss-Hermite quadrature
- robust random effects $u \sim \mathrm{Laplace}(0, \Psi)$ under the assumption $\Psi = \psi I$ $\rightarrow$ Gauss-Laguerre quadrature

Estimation of fixed effects β and covariance matrix Ψ

- gradient search for Laplace likelihood (subgradient optimization)
- derivative-free optimization (e.g., Nelder-Mead)

Numerical integration with

- normal random effects $u \sim \mathrm{N}(0, \Psi)$ $\rightarrow$ Gauss-Hermite quadrature
- robust random effects $u \sim \mathrm{Laplace}(0, \Psi)$ under the assumption $\Psi = \psi I$ $\rightarrow$ Gauss-Laguerre quadrature

Estimation of fixed effects β and covariance matrix Ψ

- gradient search for Laplace likelihood (subgradient optimization)
- derivative-free optimization (e.g., Nelder-Mead)

Numerical integration with

- normal random effects $u \sim \mathrm{N}(0, \Psi)$ $\rightarrow$ Gauss-Hermite quadrature
- robust random effects $u \sim \mathrm{Laplace}(0, \Psi)$ under the assumption $\Psi = \psi I$ $\rightarrow$ Gauss-Laguerre quadrature

Estimation of fixed effects β and covariance matrix Ψ

- gradient search for Laplace likelihood (subgradient optimization)
- derivative-free optimization (e.g., Nelder-Mead)

Numerical integration with

- normal random effects $u \sim \mathrm{N}(0, \Psi)$ $\rightarrow$ Gauss-Hermite quadrature
- robust random effects $u \sim \mathrm{Laplace}(0, \Psi)$ under the assumption $\Psi = \psi I$ $\rightarrow$ Gauss-Laguerre quadrature

Estimation of fixed effects β and covariance matrix Ψ

- gradient search for Laplace likelihood (subgradient optimization)
- derivative-free optimization (e.g., Nelder-Mead)

Numerical integration with

- normal random effects $u \sim \mathrm{N}(0, \Psi)$ $\rightarrow$ Gauss-Hermite quadrature
- robust random effects $u \sim \mathrm{Laplace}(0, \Psi)$ under the assumption $\Psi = \psi I$ $\rightarrow$ Gauss-Laguerre quadrature

Estimation of fixed effects β and covariance matrix Ψ

- gradient search for Laplace likelihood (subgradient optimization)
- derivative-free optimization (e.g., Nelder-Mead)

Numerical integration with

- normal random effects $u \sim \mathrm{N}(0, \Psi)$ $\rightarrow$ Gauss-Hermite quadrature
- robust random effects $u \sim \mathrm{Laplace}(0, \Psi)$ under the assumption $\Psi = \psi I$ $\rightarrow$ Gauss-Laguerre quadrature

Estimation of fixed effects β and covariance matrix Ψ

- gradient search for Laplace likelihood (subgradient optimization)
- derivative-free optimization (e.g., Nelder-Mead)

Numerical integration with

- normal random effects $u \sim \mathrm{N}(0, \Psi)$ $\rightarrow$ Gauss-Hermite quadrature
- robust random effects $u \sim \mathrm{Laplace}(0, \Psi)$ under the assumption $\Psi = \psi I$ $\rightarrow$ Gauss-Laguerre quadrature

Estimation of fixed effects β and covariance matrix Ψ

- gradient search for Laplace likelihood (subgradient optimization)
- derivative-free optimization (e.g., Nelder-Mead)

Numerical integration with

- normal random effects $u \sim \mathrm{N}(0, \Psi)$ $\rightarrow$ Gauss-Hermite quadrature
- robust random effects $u \sim \mathrm{Laplace}(0, \Psi)$ under the assumption $\Psi = \psi I$ $\rightarrow$ Gauss-Laguerre quadrature

Estimation of fixed effects β and covariance matrix Ψ

- gradient search for Laplace likelihood (subgradient optimization)
- derivative-free optimization (e.g., Nelder-Mead)

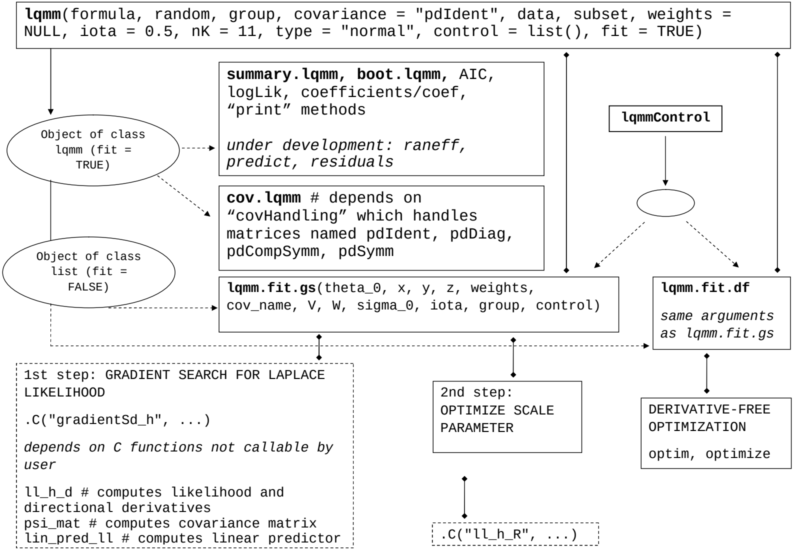
Numerical integration with

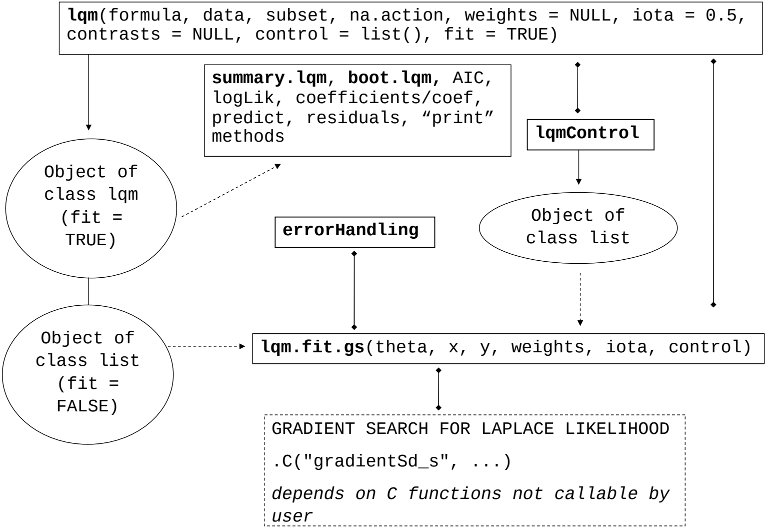
- normal random effects $u \sim \mathrm{N}(0, \Psi)$ $\rightarrow$ Gauss-Hermite quadrature
- robust random effects $u \sim \mathrm{Laplace}(0, \Psi)$ under the assumption $\Psi = \psi I$ $\rightarrow$ Gauss-Laguerre quadrature

Estimation of fixed effects β and covariance matrix Ψ

- gradient search for Laplace likelihood (subgradient optimization)
- derivative-free optimization (e.g., Nelder-Mead)

lqmm package




lqm(formula, data, subset, na.action, weights = NULL, iota = 0.5, contrasts = NULL, control = list(), fit = TRUE)
summary.lqm, boot.lqm, AIC, logLik, coefficients/coef, predict, residuals, “print” methods
lqmControl
Object of class lqm (fit = TRUE)
errorHandling
Object of class list
Object of class list (fit = FALSE)
lqm.fit.gs(theta, x, y, weights, iota, control)
GRADIENT SEARCH FOR LAPLACE LIKELIHOOD
.C("gradientSd_s", ...)
depends on C functions not callable by user

Labor pain data

- repeated measurements of self-reported amount of pain (response) on 83 women in labor
- 43 randomly assigned to a pain medication group and 40 to a placebo group
- response measured every 30 min on a 100-mm line (0 no pain - 100 extreme pain)

Aim

- to assess the effectiveness of the medication

Labor pain data

- repeated measurements of self-reported amount of pain (response) on 83 women in labor
- 43 randomly assigned to a pain medication group and 40 to a placebo group
- response measured every 30 min on a 100-mm line (0 no pain - 100 extreme pain)

Aim

- to assess the effectiveness of the medication

Labor pain data

- repeated measurements of self-reported amount of pain (response) on 83 women in labor
- 43 randomly assigned to a pain medication group and 40 to a placebo group
- response measured every 30 min on a 100-mm line (0 no pain - 100 extreme pain)

Aim

- to assess the effectiveness of the medication

Labor pain data

- repeated measurements of self-reported amount of pain (response) on 83 women in labor
- 43 randomly assigned to a pain medication group and 40 to a placebo group
- response measured every 30 min on a 100-mm line (0 no pain - 100 extreme pain)

Aim

- to assess the effectiveness of the medication

Labor pain data

- repeated measurements of self-reported amount of pain (response) on 83 women in labor
- 43 randomly assigned to a pain medication group and 40 to a placebo group
- response measured every 30 min on a 100-mm line (0 no pain - 100 extreme pain)

Aim

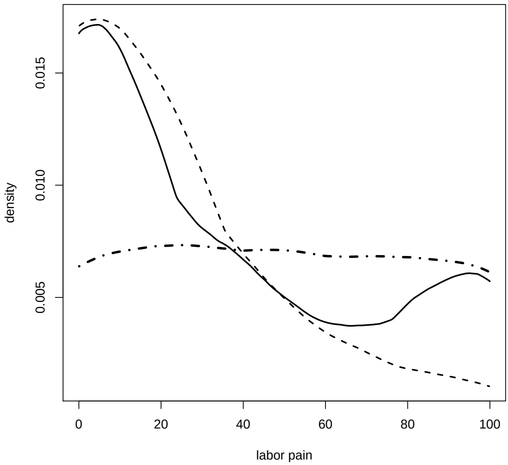
- to assess the effectiveness of the medication

Labor pain data

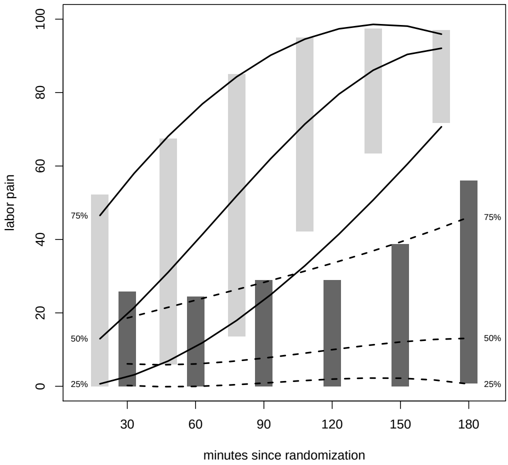
- repeated measurements of self-reported amount of pain (response) on 83 women in labor
- 43 randomly assigned to a pain medication group and 40 to a placebo group
- response measured every 30 min on a 100-mm line (0 no pain - 100 extreme pain)

Aim

- to assess the effectiveness of the medication



Density of the labor pain score plotted for the entire sample (solid line), for the pain medication group only (dashed line) and for the placebo group only (dot-dashed line). Source: Geraci and Bottai (2007).



Boxplot of labor pain score. The lines represent the estimate of the quartiles for the placebo group (solid) and the pain medication group (dashed). Source: Geraci and Bottai (2007).

```
# LQMM FIT

> tmp <- labor$y/100; tmp[tmp == 0] <- 0.025; tmp[tmp == 1] <- 0.975
> labor$pain_logit <- log(tmp/(1-tmp)) # outcome
> system.time(
+ fit.int <- lqmm(pain_logit ~ time_center*treatment, random = ~ 1, group =
labor$id, data = labor, iota = c(0.1,0.5,0.9))
+ )
   user  system elapsed
   0.22    0.00    0.22

# PRINT LQMM OBJECTS

> fit.int
Call: lqmm(formula = pain_logit ~ time_center * treatment, random = ~1,
    group = labor$id, data = labor, iota = c(0.1, 0.5, 0.9))

Fixed effects:
                       iota = 0.1  iota = 0.5  iota = 0.9
Intercept              -0.9828      0.3702      1.8547
time_center             0.7739      0.7569      0.8009
treatment              -2.6808     -2.5201     -1.8623
time_center:treatment  -0.7740     -0.7523     -0.5908

Number of observations: 357
Number of groups: 83
```

```
# EXTRACTING STATISTICS

> logLik(fit.int)
'log Lik.' -640.5730, -628.0026, -695.6520 (df=6)

> coef(fit.int)
                            0.1        0.5        0.9
Intercept             -0.9827571  0.3702043  1.8546808
time_center            0.7738773  0.7569301  0.8009082
treatment             -2.6808118 -2.5200997 -1.8623206
time_center:treatment -0.7739989 -0.7522610 -0.5907584

> cov.lqmm(fit.int)
$`0.1`
Intercept
 2.093377

$`0.5`
Intercept
 2.630906

$`0.9`
Intercept
  2.78936
```

```
# RANDOM SLOPE

> system.time(
+ fit.slope <- lqmm(pain_logit ~ time_center*treatment, random = ~ time_center,
group = labor$id, covariance = "pdSymm", data = labor, iota = c(0.1,0.5,0.9))
+ )
   user  system elapsed
   8.24    0.00    8.24
> cov.lqmm(fit.slope)
$`0.1`
            Intercept time_center
Intercept   1.4655346  0.31884671
time_center 0.3188467  0.07266096

$`0.5`
             Intercept time_center
Intercept   2.52996615 0.091073866
time_center 0.09107387 0.004447338

$`0.9`
             Intercept time_center
Intercept    1.7134076 -0.17001109
time_center -0.1700111  0.02155449
> AIC(fit.int)
[1] 1293.146 1268.005 1403.304
> AIC(fit.slope)
[1] 1375.718 1268.372 1401.859
```

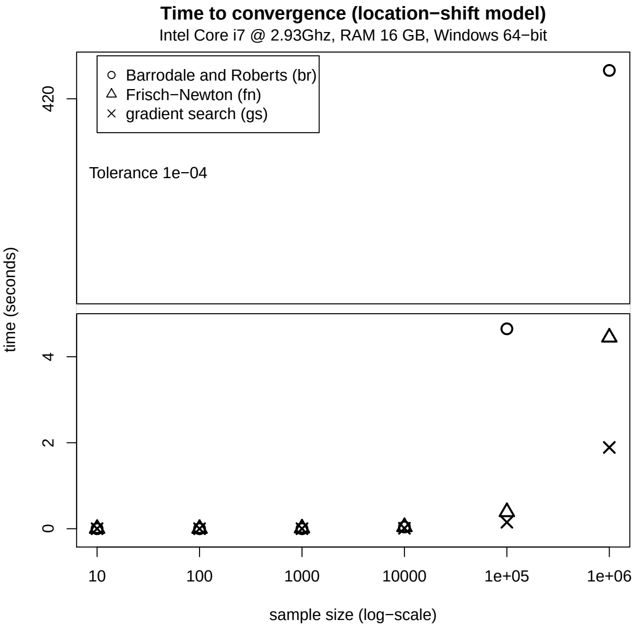
```
# SUMMARY LQMM OBJECT

> fit.int <- lqmm(pain_logit ~ time_center*treatment, random = ~ 1, group =
labor$id, data = labor, iota = 0.5, type = "robust")
> summary(fit.int)
Call: lqmm(formula = pain_logit ~ time_center * treatment, random = ~1,
    group = labor$id, data = labor, iota = 0.5, type = "robust")

Quantile 0.5
                          Value Std. Error lower bound upper bound  Pr(>|t|)
Intercept              0.011624   0.161867   -0.313661      0.3369     0.943
time_center            0.708198   0.074178    0.559132      0.8573 9.158e-13 ***
treatment             -2.989943   0.134250   -3.259729     -2.7202 < 2.2e-16 ***
time_center:treatment -0.627419   0.092040   -0.812381     -0.4425 1.275e-08 ***
scale                  0.428443   0.031258    0.365628      0.4913 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Null model (likelihood ratio):
[1] 183 (p = 0)
AIC:
[1] 1304 (df = 6)
Warning message:
In errorHandling(OPTIMIZATION$low_loop, "low", control$low_loop_max_iter,  :
  Lower loop did not converge in: lqmm. Try increasing max number of iterations
(500) or tolerance (0.001)
```

Performance assessment

- pilot simulation: confirmed previous bias and efficiency results but much faster than MCEM
- main simulation: extensive range of models and scenarios
- algorithm speed (preview):
    - lqmm method "gs" ranged from 0.03 (random intercept models) to 14 seconds (random intercept + slope) on average, for sample size between 250 ($M = 50 \times n = 5$) and 1000 ($M = 100 \times n = 10$)
    - linear programming (quantreg::rq) vs gradient search (lqmm::lqm)


Time to convergence (location−shift model)
Intel Core i7 @ 2.93Ghz, RAM 16 GB, Windows 64−bit
Barrodale and Roberts (br)
Frisch−Newton (fn)
gradient search (gs)
Tolerance 1e−04
time (seconds)
420
4
2
0
10
100
1000
10000
1e+05
1e+06
sample size (log−scale)

Concluding remarks

Work in progress

- estimation algorithms: "A Gradient Search Algorithm for Estimation of Laplace Regression" (with Prof. Matteo Bottai and Dr Nicola Orsini – Karolinska Institutet) and "Geometric Programming for Quantile Mixed Models"
- methodological: "Linear Quantile Mixed Models" (with M. Bottai) (available upon request `m.geraci@ich.ucl.ac.uk`)
- software: lqmm for Stata (with M. Bottai and N. Orsini)

Concluding remarks

To-do list (as usual, very long)

- submit to CRAN!
- plot functions
- adaptive quadrature
- integration on sparse grids (Smolyak, *Soviet Mathematics Doklady*, 1963, Heiss and Winschel, *J Econometrics*, 2008)
- missing data
- smoothing
- interface with other packages
- S4-style
- ...

Acknowledgements

Function	Package	Version	Author(s)
is./make.positive.definite	corpcor	1.5.7	J. Schaefer, R. Opgen-Rhein, K. Strimmer
permutations	gtools	2.6.2	G.R. Warnes
gauss.quad, gauss.quad.prob	statmod	1.4.8	G. Smyth

References

- Clarke F (1990). Optimization and nonsmooth analysis. SIAM.
- Geraci M, Bottai M (2007). Quantile regression for longitudinal data using the asymmetric Laplace distribution, *Biostatistics*, 8, 140-54.
- Heiss F, Winschel, V (2008). Likelihood approximation by numerical integration on sparse grids, *J Econometrics*, 144, 62-80.
- Higham N (2002). Computing the nearest correlation matrix - a problem from finance, *IMA Journal of Numerical Analysis*, 22, 329-343.
- Koenker R, Bassett G (1978). Regression quantiles, Econometrica, 46, 33-50.
- Machado JAF, Santos Silva JMC (2004). Quantiles for counts, *JASA*, 100, 1226-1237.
- Smolyak S. (1963). Quadrature and interpolation formulas for tensor products of certain classes of functions, *Soviet Mathematics Doklady*, 4, 240-243.

MRC Centre of Epidemiology for Child Health
Institute of Child Health, University College London
http://www.ucl.ac.uk/ich/homepage

m.geraci@ich.ucl.ac.uk