Online Resource 1

1 Introduction

This report contains supplementary material for the paper entitled "Multiple imputation for bounded variables" (hereafter, the Manuscript). This material consists of sections with details on computation and software.

2 Nonlinear estimation

For fitting model (4) based on transformation (5) as defined in the Manuscript, Geraci and Jones (2015) discuss two-stage (TS) estimation (Chamberlain, 1994; Buchinsky, 1995; Fitzenberger et al., 2010), residual cusum process (RC) estimation (Mu and He, 2007), interior point algorithms for nonlinear quantile regression (Koenker and Park, 1996), and derivative-free optimization for non-smooth functions (Nelder and Mead, 1965). In simulation studies, they found that all estimators were consistent in terms of bias but had different computational performance. The least attractive of all was the RC estimator, with a 30:1 computing time ratio relative to the TS estimator. However, even the relatively fast TS estimator would entail prohibitive computing times when sampling as little as 5 imputations for a modest number of incomplete observations. Moreover, since the TS estimator searches the optimal value of λ over a grid of pre-specified values (i.e., by means of profiling), it would be particularly challenging to find a grid that is coarse and narrow enough to speed up computation, but sufficiently wide to ensure the optimal value is included.

Nonlinear estimation is a fast, viable alternative, though occasionally it may lack numerical stability. The goal is to jointly estimate $\boldsymbol{\beta}_p$ and λ_p by fitting the nonlinear model in Equation (6) of the Manuscript, that is

$$Q_{g(Z)|X}(p) = \left[\lambda_p(\alpha_p + \mathbf{x}^\top\boldsymbol{\beta}_p) + \sqrt{1 + \{\lambda_p(\alpha_p + \mathbf{x}^\top\boldsymbol{\beta}_p)\}^2}\right]^{1/\lambda_p},$$

where the variable $g(Z)$ is related to Z according to Table 1 of the Manuscript. (Note that the model above is defined for $\lambda_p \neq 0$ and that $Q_{g(Z)|X}(p) \approx \exp\left(\mathbf{x}^\top \boldsymbol{\beta}_p\right)$ when $|\lambda_p| \to 0$.) In preliminary numerical investigations, we first considered an interior point algorithm (Koenker and Park, 1996). Unfortunately, the imputation procedure would often halt before completion due to the algorithm's failure to converge. In contrast, estimation based on an adaptation of a gradient search algorithm (Geraci and Bottai, 2014; Bottai et al., 2015) proved to be feasible and appropriate for our purpose. These methods are implemented in the R (R Core Team, 2016) package `Qtools` (Geraci, 2016, 2017), for which a sample code is offered in the next section. In the following, we briefly describe the algorithm.

Let z_i, $i = 1, \ldots, n$, be a sample of observations of a singly or doubly bounded continuous variable Z and let $\mathbf{x}_i$ be a $(q-1) \times 1$ vector of predictors for Z. On the scale of h, we assume the linear model (4) in the Manuscript. We define the objective function

$$f(\boldsymbol{\theta}) = \sum_{i=1}^{n} r_i \left(p - I_{r_i<0}\right),$$

where $\boldsymbol{\theta} = \left(\lambda_p, \alpha_p, \boldsymbol{\beta}_p^\top\right)^\top$ is the $(q+1) \times 1$ parameter to be estimated, $r_i = z_i - h^{-1}\left(\alpha_p + \mathbf{x}_i^\top \boldsymbol{\beta}_p; \lambda_p\right)$, and $I_A = 1$ if A is true or $I_A = 0$ otherwise. We also define the (directional) gradient

$$\dot{f}(\boldsymbol{\theta}) \equiv \frac{\partial f(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \begin{pmatrix} -\sum_i \dfrac{\partial h^{-1}\left(\alpha_p + \mathbf{x}_i^\top \boldsymbol{\beta}_p; \lambda\right)}{\partial \lambda} \left(p - I_{r_i<0}\right) \\ -\sum_i \mathbf{x}_i \dfrac{\partial h^{-1}\left(\eta; \lambda_p\right)}{\partial \eta} \left(p - I_{r_i<0}\right) \end{pmatrix}.$$

Note that $f(\boldsymbol{\theta})$ is Lipschitz near $\boldsymbol{\theta}$. See Geraci and Bottai (2014) for a discussion on nonsmooth optimisation of Lipschitz functions using Clarke's derivatives (Clarke, 1990).

From a current parameter value the algorithm searches the positive semiline in the direction of the gradient for a new parameter value at which the objective is smaller. The algorithm stops when the change in the objective is less than a specified tolerance. Convergence is guaranteed by the continuity and concavity of the L_1-norm loss function (Bottai et al., 2015). Let $\boldsymbol{\theta}^{(t)}$ denote the value of the parameter at iteration t. The minimization steps are:

(1) Set $\boldsymbol{\theta} = \boldsymbol{\theta}^{(t)}$; $\delta = \delta^{(t)}$; $t = 0$.

```
(2) Define $\boldsymbol{\theta}^{*} = \boldsymbol{\theta}^{(t)} - \delta^{(t)} \dot{f}\left(\boldsymbol{\theta}^{(t)}\right)$. If $f\left(\boldsymbol{\theta}^{*}\right) > f\left(\boldsymbol{\theta}^{(t)}\right)$
    (a) then set $\delta^{(t+1)} = a\delta^{(t)}$;
    (b) else if $\left|f\left(\boldsymbol{\theta}^{*}\right)/f\left(\boldsymbol{\theta}^{(t)}\right) - 1\right| < \omega$
        (i) then return $\boldsymbol{\theta}^{*}$; stop;
        (ii) else set $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{*}$; $\delta^{(t+1)} = b\delta^{(t)}$.
(3) Set $t = t + 1$; go to step 2.
```

The algorithm requires setting the starting value of the parameter $\boldsymbol{\theta}^{0}$, the initial step $\delta^{(0)} > 0$, the contraction step factor $a \in (0, 1)$, the expansion step factor $b \geq 1$, and the tolerance $\omega > 0$ for the change in f.

3 R code

In this section, we provide sample R code to show how to apply our proposed multiple imputation method using the A-level Chemistry Scores dataset. The dataset contains observations on A-level scores in Chemistry for 31,022 English and Welsh students and is available from the `Qtools` package.

```
# Load packages
library(mice)
library(quantreg)
library(Qtools)

# Load dataset Chemistry from package Qtools
data(Chemistry)

# A-level scores in Chemistry (range 0-10) for 31022 students
summary(Chemistry$score)

# See more details on dataset in R documentation
?Chemistry
```

The A-level score is a variable bounded between 0 and 10. The data set is completely observed. For illustration purposes, we randomly generate about 20% missing values in the score variable as well as in the age variable.

```
# Randomly generate missing values (about 20%)
set.seed(123)
n <- nrow(Chemistry)
M <- rbinom(n, 1, .2)
Chemistry$score[M == 1] <- NA
M <- rbinom(n, 1, .2)
Chemistry$age[M == 1] <- NA
```

Next, we create a matrix of variables that feed into `mice`. The matrix `X` contains the variables `score` and `age`, which are to be imputed, and the variables `sex`, which is completely observed and is included as auxiliary variable to predict the missing data. Note that age is expressed in months.

```
# Define matrix for mice
X <- Chemistry[,c("score","age","sex")]
```

The function `mice` provides a number of arguments specific to its algorithm, such as the number of imputations (`m`), the number of Gibbs sampler's iterations (`maxit`) and the imputation model for each of the variables with missing values (`method`). The argument `method` is a vector of the same length as the number of variables in the dataset that defines the imputation model for each variable in the dataset (but it can be also a single string vector or be empty – see `?mice` for further details). Different methods are available from `mice`, including predictive mean matching (`pmm`), (Bayesian) linear regression (`norm`), and logistic regression (`logreg`). Quantile-regression imputation (`rq`) is available from `Qtools`. In this example, we choose quantile-regression imputation for the bounded variable (`score`) and, say, linear regression for `age`. Since the third variable need not be imputed, we can simply use the empty method. The syntax is thus `c("rq", "norm", "")`.

There are a number of arguments relating to quantile-regression imputation and these are documented in the `Qtools` package (see `?mice.impute.rq`). For example, the argument `tsf` specifies the transformation to be used; `dbounded` indicates whether the variable is doubly bounded or not, and, if doubly bounded, the bounds are given in `x.r` (if not given, the bounds are defined as the observed minimum and maximum); `conditional` is a logical flag to indicate if the transformation parameter is assumed to be known, in which case it must be provided via the argument `lambda` (otherwise it is estimated from the data using the algorithm described in Section 2 above).

```
# Impute score using transformation-based quantile regression
imp <- mice(X, method = c("rq", "norm", ""), m = 5, maxit = 5, x.r
= c(0,10), tsf = "mcjI", dbounded = TRUE, conditional = TRUE, lambda
= 0)
```

The chained equations resulting from the instruction above will be

$$\begin{aligned} Q_{\text{score}|\text{age},\text{sex}}(p) &= h^{-1}\left\{Q_{h(Z;0)|\text{age},\text{sex}}(p);0\right\}, \\ \mathrm{E}(\text{age}|\text{score},\text{sex}) &= \alpha + \mathbf{x}^{\top}\boldsymbol{\beta}. \end{aligned}$$

Once the algorithm terminates, the imputations can be used to calculate the statistics of interest. In the example below, we consider the model $\text{logit}(\pi) = \theta_0 + \theta_1\text{sex} + \theta_2\text{age}$, where $\pi = \Pr\left(I_{\texttt{score}>8}\right)$. This will produce as many estimates of the regression coefficients as the number of imputations (i.e., five). The estimates are then ‘pooled’ together with the function `pool` to produce one single summary.

```
# Estimate logistic regression
fit <- with(data = imp, exp = glm(I(score > 8) ~ sex + age,
family = binomial()))

# Pool results
pool(fit)
```

References

Bottai, M., N. Orsini, and M. Geraci (2015). A gradient search maximization algorithm for the asymmetric Laplace likelihood. *Journal of Statistical Computation and Simulation 85*(10), 1919–1925. doi:10.1080/00949655.2014.908879.

Buchinsky, M. (1995). Quantile regression, Box-Cox transformation model, and the US wage structure, 1963-1987. *Journal of Econometrics 65*(1), 109–154. doi:10.1016/0304-4076(94)01599-U.

Chamberlain, G. (1994). Quantile regression, censoring, and the structure of wages. In C. Sims (Ed.), *Advances in Econometrics: Sixth World Congress*, Volume 1. Cambridge, UK: Cambridge University Press.

Clarke, F. H. (1990). *Optimization and nonsmooth analysis.* Society for Industrial Mathematics.

Fitzenberger, B., R. A. Wilke, and X. Zhang (2010). Implementing Box–Cox quantile regression. *Econometric Reviews 29*(2), 158–181. doi:10.1080/07474930903382166.

Geraci, M. (2016). Qtools: A collection of models and tools for quantile inference. *The R Journal 8*(2), 117–138.

Geraci, M. (2017). *Qtools: Utilities for Quantiles.* R package version 1.2. URL: https://CRAN.R-project.org/package=Qtools.

Geraci, M. and M. Bottai (2014). Linear quantile mixed models. *Statistics and Computing 24*(3), 461–479. doi:10.1007/s11222-013-9381-9.

Geraci, M. and M. C. Jones (2015). Improved transformation-based quantile regression. *Canadian Journal of Statistics 43*(1), 118–132. doi:10.1002/cjs.11240.

Koenker, R. and B. J. Park (1996). An interior point algorithm for nonlinear quantile regression. *Journal of Econometrics 71*(12), 265–283. doi:10.1016/0304-4076(96)84507-6.

Mu, Y. M. and X. M. He (2007). Power transformation toward a linear regression quantile. *Journal of the American Statistical Association 102*(477), 269–279. doi:10.1198/016214506000001095.

Nelder, J. A. and R. Mead (1965). A simplex method for function minimization. *Computer Journal 7*(4), 308–313. doi:10.1093/comjnl/7.4.308.

R Core Team (2016). *R: A language and environment for statistical computing.* Vienna, Austria: R Foundation for Statistical Computing.