cbqr

In this file, we detail the documentation of the functions in `cbqr` implemented in `R`.

❊ cbqr.gs

Description

`cbqr.gs` implements the methodology censored bent line quantile regression. In the first step, it estimates an informative subset (Tang, Wang, He, & Zhu, 2012). In the second step, it estimates a bent line quantile regression model based on (Li, Wei, Chappell, & He, 2011). The estimation of the change points is based on grid search, and the estimation of regression coefficients is based on `rq` in the package `quantreg`. The standard errors and the interval estimates are provided.

Usage

```
cbqr.gs(cy, tt, xx, tau, delta, level, censoring = c('left', 'right'), method = c('ks',
'nid', 'ker'))
```

Arguments

- `cy`: the response (including the censored response).
- `tt`: the covariate that may contain a change point.
- `xx`: the covariates that do not contain a change point.
- `tau`: quantile level
- `delta`: the censoring indicator (0 if censored).
- `level`: significance level. The default is 5%.
- `censoring`: the censoring type. `'right'` means censoring from above while `'left'` means censoring from below. The default is `'right'`.
- `method`: the methods for density estimation involved in asymptotic interval estimation. See Details. The default is `'ks'`.

Details

Similar to Li et al. (2011), we assume the following model for the latent response variable $Y_{i,j}^*$,

$$Y_{i,j}^* = \begin{cases} \alpha_{i,1} + \beta_1 x_{i,j} + z_{i,j}^\top \gamma + \varepsilon_{i,j} & \text{if } x_{i,j} \le t \\ \alpha_{i,2} + \beta_2 x_{i,j} + z_{i,j}^\top \gamma + \varepsilon_{i,j} & \text{if } x_{i,j} > t \end{cases}, \quad i = 1, \ldots, n, j = 1, \ldots, m_i.$$

Define $N = \sum_i m_i$. We only observe $Y_{i,j} = \max\{Y_{i,j}^*, C\}$, where C is a known censoring time point. Following the notations of Li et al. (2011), we are interested in estimating the τth conditional quantile function as

$$Q(\boldsymbol{w}_{i,j}\,;\,\boldsymbol{\theta}_0) = \alpha_i + \Big[\beta_1 I(x_{i,j} \le t) + \beta_2 I(x_{i,j} > t)\Big](x_{i,j} - t) + \boldsymbol{z}_{i,j}^\top \boldsymbol{\gamma},$$

where $\boldsymbol{w}_{i,j} = (I(i=1), \ldots, I(i=n), x_{i,j}, \boldsymbol{z}_{i,j}^\top)^\top$ and $\boldsymbol{\theta}_0 = (\alpha_1, \ldots, \alpha_n, \beta_1, \beta_2, \boldsymbol{\gamma}^\top)^\top$. Therefore, for some small positive constant c_N, the estimation of $\boldsymbol{\theta}_0$ is based on

$$\hat{\boldsymbol{\theta}}_N = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{n} \sum_{j=1}^{m_i} \rho_\tau \big[Y_{i,j} - g(\boldsymbol{w}_{i,j}\,;\,\boldsymbol{\theta})\big] I\big[\hat{\pi}(\boldsymbol{w}_{i,j}) > \tau\big]$$

where the estimate of the informative subset $\hat{\pi}(\boldsymbol{w}_{i,j})$ is obtained by using generalized additive models (GAM) with covariates $\{\boldsymbol{w}_{i,j} : i = 1, \ldots, n, j = 1, \ldots, m_i\}$.

`method = 'ks'` implements a kernel density estimation, where the residuals are corrected for bias according to the derived Bahadur representation of the estimator. `method = 'ker'` is recommended for larger sample sizes as it may occasionally produce wide intervals when the sample size is small. See details of `method = 'ker'` and `method = 'nid'` in `summary.rq`.

Value

- `beta.est`: estimates of β_1, β_2 along with standard error estimates.
- `t.est`: estimates of t along with standard error estimates.
- `ci.t`: confidence intervals of t_0
- `ci.beta`: confidence intervals of β_1, β_2
- `gamma.est`: estimates of γ along with standard error estimates.

Author

Jilei Lin (`jileil2@gwmail.gwu.edu`).

References

Li, C., Wei, Y., Chappell, R., & He, X. (2011). Bent line quantile regression with application to an allometric study of land mammals' speed and mass. *Biometrics*, *67*(1), 242–249.

Tang, Y., Wang, H. J., He, X., & Zhu, Z. (2012). An informative subset-based estimator for censored quantile regression. *Test*, *21*, 635–655.

Examples

```
subj.m <- seq(from = .5, to = 3, length.out = m)
yy <- c()
tt <- c()
cy <- c()
for (j in 1:m) {
  t <- runif(n, 0, 3)
  t <- t[order(t)]
  tt <- c(tt, t)
  y <- subj.m[j] + .5 * t + 2 * pmax(t - t0, 0) + rnorm(n, sd = .1)

  # censoring
  c <- y
  if (min(which(y > cl)) < Inf) {
    c[min(which(y > cl)):length(c)] <- cl
  }

  yy <- c(yy, y)
  cy <- c(cy, c)
}

# covariate
group <- as.factor(sapply(1:m, function(x) rep(x, n)))
xx <- as.matrix(model.matrix(~ group))
xx <- xx[, - 1]

###################### Estimated ISUB

## GAM estimation
delta <- ifelse(yy <= cl, yes = 1, no = 0)
```

```
## Bent QR with estimated ISUB (grid search)
mod.gs <- cbqr.gs(cy = cy, tt = tt, xx = xx, tau = tau, delta = delta)
```