Functional Censored Quantile Regression

Fei Jiang and Qing Cheng

1 Introduction

This vignette provides an introduction to the `FCQR` package. R package `FCQR` implements method for Functional Censored Quantile Regression.

```
remotes::install_github("QingCheng0218/FCQR@main");
```

Load the package using the following command:

```r
library(FCQR);
```

2 Simulation Study

2.1 Finite Sample Performance Without IGACV Procedure

In this section, we conduct simulation studies to assess the finite-sample performance of the FCQR procedure. We first load the dependent R packages as follows:

```r
library(xtable); library(survival); library(fda);
library(quantreg); library(ggplot2); library(MASS);
```

We generate the event time from the model,

$$\log(T_i) = b_1 X_{1i} + \int_0^1 Z_i(s)\phi(s)ds + \left\{ b_2 X_{2i} + \int_0^1 Z_i(s)ds \right\} \epsilon_i,$$

where X_{1i} is a normal variate with variance σ_z^2, X_{2i} is a variate uniformly distribution on $[0, c_0]$, and ϵ_i is a normal random error with mean zero and variance σ_ϵ^2. The detailed information of functional covariate $Z_i(s)$ and $\phi(s)$ can be found in Section 5.1 of our paper. We set $\sigma_\epsilon = 0.2$, $\nu = 2.5$, $n = 200$ and λ_0 is used to control the censored rate.

```r
n <- 200; v <- 1; sdx <- 0.2; sdz <- 1; lam0 <- 5;

t <- seq(0, 1, 0.01);  m <- length(t); deltat <- (diff(t)[1]);
nk <- 50; c0 <- 1;
b0 <- c(1, 2); lb0 = length(b0);

grd <- seq(0.1, 0.9, 0.1/5);
qv <- c(which(eleq(0.3, grd)), which(eleq(0.4, grd)), which(eleq(0.5, grd)),
        which(eleq(0.6, grd)), which(eleq(0.7, grd)));
```

We using the following function to B-spline bases.

```r
# --------------------------------------------------------#
# creat B-spline basis.
nbasis <- 5; norder <- 4;
basis <- create.bspline.basis(range(t),nbasis = nbasis, norder = norder)
nbasis <- basis$nbasis
```

```r
valueBasis <-eval.basis(t, basis)
# ---------------------------------------------------------#
```

```r
nsim = 10;
Base0.m <- Sigb0.m <- array(0, dim = c(nsim, length(qv), lb0));
Bfun.m <- Sigfun.m <- array(0, dim = c(nsim, m, length(qv)))
cdata <- matrix(0, ncol = nsim, nrow = n);
itr <- 0; tt <- 1;
repeat{
  itr <- itr +1
  # ---------------------------------------------------------#
  # Generate Sim function.
  temp <- simfun(t, v, n, sdx, sdz, b0,  lam0, c0);
  Zb <- temp$Zb # baseline covariate.
  time <- temp$Timedata # survival time and censoring time.
  Zs <- temp$mZ # functional covariate.
  cdata[, itr] <- 1*(time[, 1] <= time[, 2])
  # ---------------------------------------------------------#
  res.itr <- try(Fcqr(time, Zs, Zb, t, grd, qv, valueBasis));
  if(class(res.itr) != 'try-error'){
    Base0.m[itr, , 1:lb0] <- res.itr$base0;
    Sigb0.m[itr, ,1:lb0] <- res.itr$sigb0;
    Bfun.m[itr, , ] <- t(res.itr$Bfun);
    Sigfun.m[itr, , ] <- t(res.itr$sigfun);
    tt <- tt + 1;
  }
  if(tt%%100==0) {print(tt);print(itr);print(Sys.time())}
  if(tt == nsim + 1){
    break
  }
}

cenRate = 1 - mean(colMeans(cdata))
```

```r
#**********************************#
# The result of coefficient function.
#**********************************#
alfun <- simBX(t, v, nk)$B0
plot.m <- alfun.be(alfun, Sigfun.m, Bfun.m, grd, qv, sdx);
tau.id <- 3;
resplot6 <- alphaplot.be(t, grd, qv, tau.id, plot.m, -10, 1);
```

```r
#**********************************#
# The result of Beta0
#**********************************#
lgrd <- length(grd);
btautrue <- cbind(rep(b0[1], lgrd), b0[1]*qnorm(grd, 0, sdx))
b0res <- rbind(b0res.be(Base0.m[,,1], Sigb0.m[,,1], btautrue[,1], qv),
            b0res.be(Base0.m[,,2], Sigb0.m[,,2], btautrue[,2], qv));
colnames(b0res) = c("BIAS", "SD", "SE", "CP");
rownames(b0res) = paste0(rep(c("$\\hat\\beta_1(","$\\hat\\beta_2("), 5),
                         rep(grd[qv], each  = 2),rep(")$", 10));
```

Table 1 shows the estimation result for $\beta(\tau)$, the result of time-varying estimator can be found in Figure 1. One can increase the number of replications, denoted by *nsim*, to obtain more accurate results.

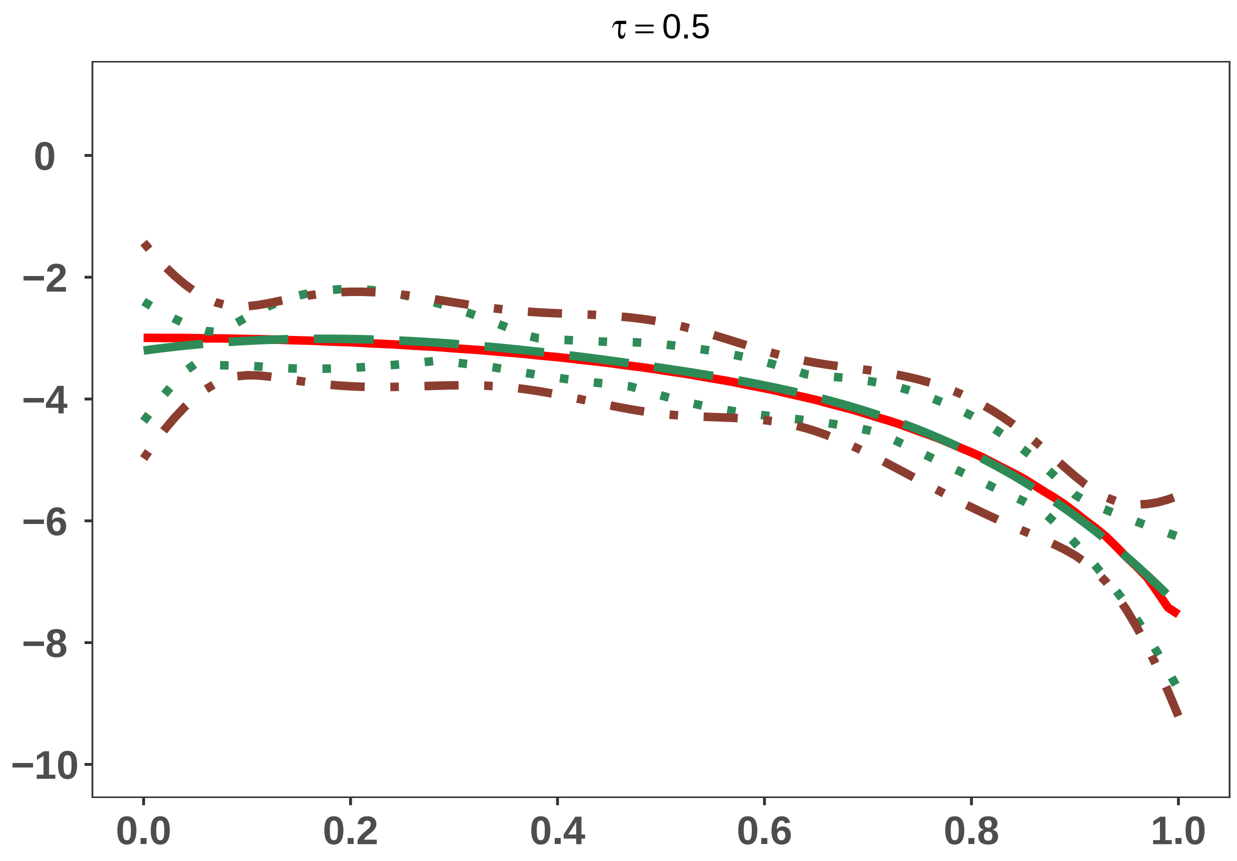

Figure 1: Estimated coefficient function $B_{rS}^T(s)\hat{\gamma}_S(0.5)$ under a censoring rate of 20%.

Table 1: Estimation results for $\hat{\beta}(\tau) = (\hat{\beta}_1(\tau_1), \hat{\beta}_2(\tau_2)$ at different quantiles with a censoring rate of 20%.

	BIAS	SD	SE	CP
$\hat{\beta}_1(0.3)$	0.0071215	0.0549538	0.0613021	1.0
$\hat{\beta}_2(0.3)$	0.0061289	0.0430495	0.0630073	1.0
$\hat{\beta}_1(0.4)$	0.0125346	0.0553038	0.0602159	0.9
$\hat{\beta}_2(0.4)$	0.0164120	0.0756839	0.0690479	0.9
$\hat{\beta}_1(0.5)$	0.0191060	0.0697755	0.0713653	0.9
$\hat{\beta}_2(0.5)$	0.0916022	0.1914818	0.1854501	0.8
$\hat{\beta}_1(0.6)$	0.0632341	0.2052910	0.2063237	0.9
$\hat{\beta}_2(0.6)$	0.0200254	0.1385350	0.1899308	1.0
$\hat{\beta}_1(0.7)$	0.0277758	0.1610413	0.2220223	1.0
$\hat{\beta}_2(0.7)$	0.1168473	0.1931968	0.2404025	1.0

```
resplot6
```

2.2 Performance of IGACV for Knot Selection

In this section, we mimic the real data to illustrate the performance of the IGACV procedure.

```r
rm(list = ls())
library(FCQR);
library(xtable); library(survival); library(fda);
library(quantreg); library(ggplot2); library(MASS);
```

In particular, we generate $n = 297$ survival time from the model

$$\log(T_i) = \int_0^1 Z_i(s)\phi(s)ds + \left\{b_1 X_{2i} + \int_0^1 Z_i(s)ds\right\}\epsilon_i,$$

where $Z_i(s) = SBP_i(s) + U_i(s)$ and $Z_{1i} = \min(SBP_i) + U_{1i}$ with $SBP_i(s)$ corresponding to the systolic blood pressure(SBP) trajectory for the $i-$th individual, and $U_i(s)$ and U_{1i} are uniform random variables on $[-c_0, c_0]$.

```r
# ------------------------------------------------------------
t <- time; zsm <- MSdata[, 3:98];
c0 <- 10;  sdx <- 0.2; sdz <- 1; beta1 <- 0.1; norder <- 4;
# ------------------------------------------------------------
grd <- seq(0.1, 0.9, 0.005/5);
qv <- c(which(eleq(0.3, grd)), which(eleq(0.4, grd)), which(eleq(0.5, grd)),
        which(eleq(0.6, grd)), which(eleq(0.7, grd)));
# ------------------------------------------------------------
```

We evaluate IGACV and IBIC over the choices of $d_s = 4, 5, 6, 7, 8$.

```r
bsnb <- c(4, 5, 6, 7, 8);
nsim <- 10; lam0 <- 1 # for 20% censoring rate.
resop = optbasis(nsim, bsnb, zsm, lam0, c0, beta1, sdz, sdx,
                 grd, t, norder, qv);
resoptb = resop$inopt;
rownames(resoptb) = c("IGACV", "IBIC")
colnames(resoptb) = bsnb;
```

Table 2: Optimal d_S based on the IGACV and IBIC criteria under different censoring rate of 20% over 10 simulations.

	4	5	6	7	8
IGACV	1	2	6	1	0
IBIC	4	3	3	0	0

IGACV chooses $d_S = 6$ as the optimal number of basis functions, see Table 2. Next, we present the estimation results for d_S in Table 3 and Figure 2.

```r
# ------------------------------------------------------------
nbasis <- 6;
btautrue <- beta1*qnorm(grd, 0, sdx);
# ------------------------------------------------------------
# report tau equals 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8;
grd <- seq(0.1, 0.9, 0.1/5);
qv <- c(which(eleq(0.2, grd)),
        which(eleq(0.3, grd)), which(eleq(0.4, grd)),
        which(eleq(0.5, grd)), which(eleq(0.6, grd)),
        which(eleq(0.7, grd)), which(eleq(0.8, grd)));
```

```r
# ------------------------------------------------------------
SIMRES <- Mimicfun(zsm, lam0, c0, beta1, sdz, sdx, nsim,
                   grd, t, nbasis, norder, qv);

Base0.m <- SIMRES$Base0.m;
Sigb0.m <- SIMRES$Sigb0.m;
Sigfun.m <- SIMRES$Sigfun.m;
Bfun.m <- SIMRES$Bfun.m;

b0res <- b0res.be(Base0.m, Sigb0.m, btautrue, qv)
colnames(b0res) = c("BIAS", "SD", "SE", "CP");
rownames(b0res) = seq(0.2, 0.8, 0.1);

n <- nrow(MSdata);
alfun <- SimMimicfun(1, zsm, t, n, sdz, sdx, c0, beta1, lam0)$alp;
plotres <- alfun.be(alfun, Sigfun.m, Bfun.m, grd, qv, sdx);
tau.id <- 4;
mimicplot <- alphaplot.be(t, grd, qv, tau.id, plotres, -16, 16);
```

Table 3: Estimation results for $\hat{\beta}(\tau)$ at different quantiles with a censoring rate of 20%.

	BIAS	SD	SE	CP
0.2	0.0268858	0.1566130	0.2376471	1.0
0.3	0.0627040	0.1999660	0.2025017	0.9
0.4	0.0033346	0.1957361	0.2056496	0.9
0.5	0.0189750	0.1472664	0.2331903	1.0
0.6	0.0470778	0.1696225	0.2111539	1.0
0.7	0.0199307	0.1474049	0.2453912	1.0
0.8	0.0471585	0.1688762	0.2524428	1.0

```r
mimicplot;
```

Again, the accuracy of results can be improved by increasing the number of simulations.

3 Stroke Application

In this section, we identify functional relationship between ambulatory blood pressure trajectories and clinical outcomes in stroke patients. We first apply the IGACV and IBIC criteria to select d_S.

```r
rm(list=ls());
library(FCQR);
# ---------------------------------------------------
library(imputeTS); library(survival); library(survminer); library(fda);
library(quantreg); library(statmod); library(xtable); library(Cairo);
# ---------------------------------------------------

a0 <- 0; b0 <- 1; gn <- 20;
norder <- 4;
grd <- seq(0.1, 0.9, 0.005/5); lgrd <- length(grd);
qv <- c(which(eleq(0.2, grd)),
```

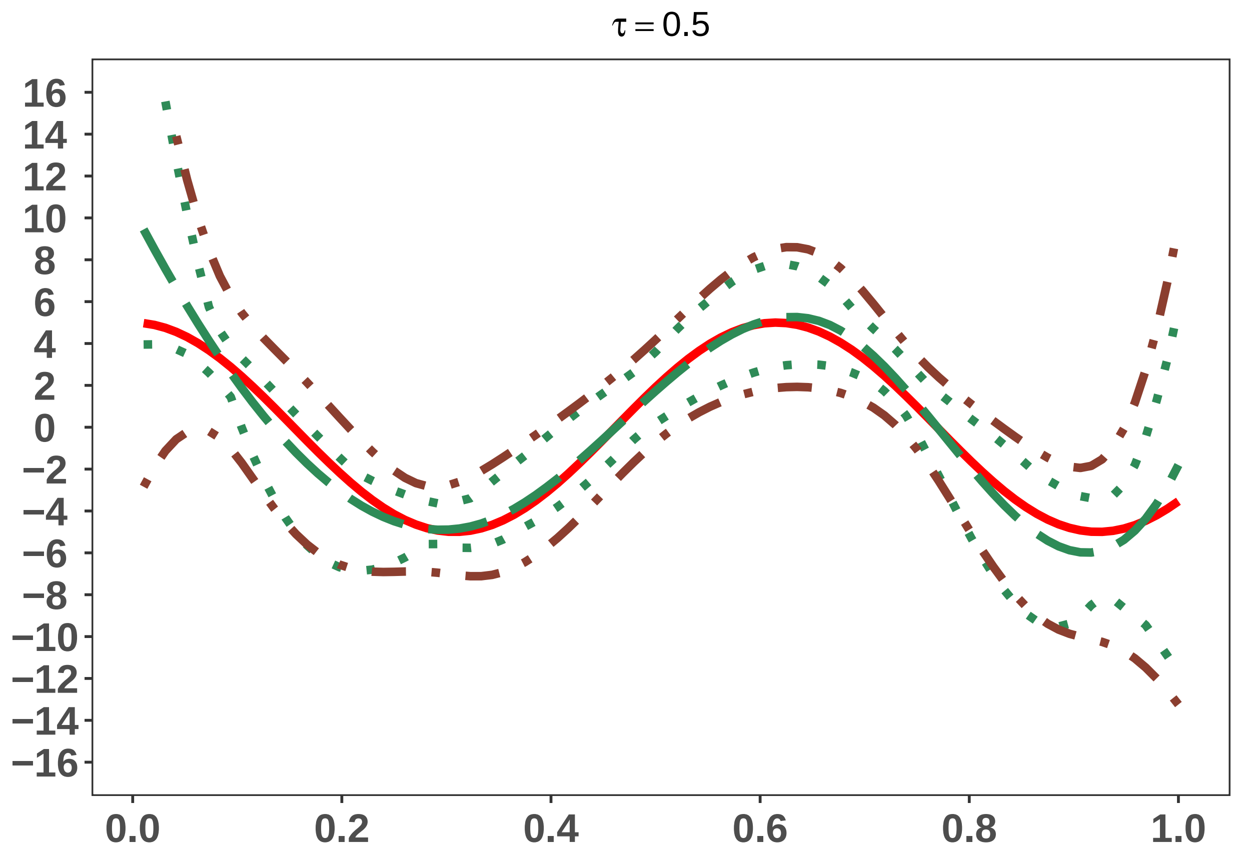

Figure 2: Estimated coefficient function $B_{rS}^T(s)\hat{\gamma}_S(0.5)$ with $d_S = 6$ under a censoring rate of 20%.

```
        which(eleq(0.3, grd)), which(eleq(0.4, grd)), which(eleq(0.5, grd)),
        which(eleq(0.6, grd)), which(eleq(0.7, grd)), which(eleq(0.8, grd)));
# -----------------------------------------------------#
# using GACV method to determine optimal nbasis
CBA <- c(1, 2, 3, 4, 5)
BICR <- GACVR <- matrix(0, ncol=length(CBA), nrow=lgrd)
NBS <- rep(0,length(CBA))

n <- dim(MSdata)[1]
cn <- (log(n))^(1/2)
for(i in 1:length(CBA)){
  cba <- CBA[i];
  res <- predfun(MSdata, a0, b0, gn, time, norder, cba);
  TM <- res$TM; DEL <- res$DEL;
  nbasis <- res$nbasis; ns <- nbasis + 1;
  pre.y <- res$pre.y;
  bicres <- bicfun(n, ns, cn, TM, pre.y, DEL, grd);
  GACVR[, i] <- bicres$gacv;
  BICR[, i] <- bicres$bic;
  NBS[i] <- nbasis;
}

bicres <- optfun(grd, BICR);
gacvres <- optfun(grd, GACVR);
resoptReal <- rbind(gacvres[2, ], bicres[2, ]);
rownames(resoptReal) <- c("IGACV", "IBIC");
colnames(resoptReal) <- 4:8;
# -----------------------------------------------------#
```

Table 4: Selection of the optimal number of basis functions using the IGACV and IBIC criteria respectively for the stroke data.

	4	5	6	7	8
IGACV	0.35430	0.35546	0.35352	0.35327	0.35552
IBIC	-0.61647	-0.60538	-0.60105	-0.59384	-0.58068

As shown in Table 4, the IGACV reaches the minimum value at $d_S = 7$. Based on $d_S = 7$, we estimate $\beta(\tau)$ and $\alpha(s, \tau)$ using the following command.

```
cba <- 4;
covres <- intzfun0(MSdata, a0, b0, gn, time, norder, cba)
covdata <- covres$covdata
Z <- covres$Z
valueBasisT <- covres$valueBasisT
reswei <- weifun(MSdata, covdata)
# weight0 <- reswei
weight0 <- rep(1, nrow(MSdata))
coefres <- coeffun0(valueBasisT, covdata,  MSdata,
                    grd, qv, weight0, Z, time)
# ----------------------------------------------------
# Result of baseline covariate
base0 <- coefres$base0
sigb0 <- coefres$sigb0
```

```r
c2 <- base0 + 1.96 * sqrt(sigb0);
c1 <- base0 - 1.96 * sqrt(sigb0);

ci0 <- paste("(",round(c1,5)," ,",round(c2,5),")",sep="")
resbase0 <- rbind(round(base0, 5), ci0)
rownames(resbase0) <- c("Estimators", "95% CI ")
resbase = resbase0[, 2:5];
colnames(resbase) = seq(0.3, 0.6, 0.1);
```

Table 5: Estimates $\hat{\beta}(\tau)$ and the 95% bootstrap confidence intervals (CIs) at different values of τ for the stroke data.

	0.3	0.4	0.5	0.6
Estimators	-0.00859	-0.00735	-0.00725	-0.00681
95% CI	(-0.01192 ,-0.00527)	(-0.00974 ,-0.00495)	(-0.00933 ,-0.00517)	(-0.00808 ,-0.00554)

```r
# Result of time dependent covariate
Bfun <- coefres$Bfun
sigfun <- coefres$sigfun
DA00 <- DA01 <- DA02 <- matrix(0, ncol = length(time), nrow = length(grd[qv]))
for(j in 1:length(grd[qv])){
  DA00 <- Bfun;
  bt <- DA00[j, ];
  vc <- sigfun[j, ];
  DA01[j, ] <- bt + 1.96 * sqrt(vc);
  DA02[j, ] <- bt - 1.96 * sqrt(vc);
}

tau.id <- 3;
bt <- Bfun[tau.id, ];
vc <- sigfun[tau.id, ];
da00 <- bt;
da01 <- bt + 1.96 * sqrt(vc);
da02 <- bt - 1.96 * sqrt(vc)

yid <- 0.5; max.y <- yid; min.y <- -yid;
t <- seq(19.25, 43, length.out = 96);
realplot = bandplot(da00, da01, da02, tau.id, t, min.y, max.y);
```

```r
realplot;
```

Table 5 shows that there are significant negative effects of Z_{1i} at all considered quantiles, which suggests that hypertension is an important risk factor for stroke. Figure 3 displays the estimated $\alpha(s, \tau)$ and the corresponding 95% pointwise bootstrap confidence intervals with $\tau = 0.4$.

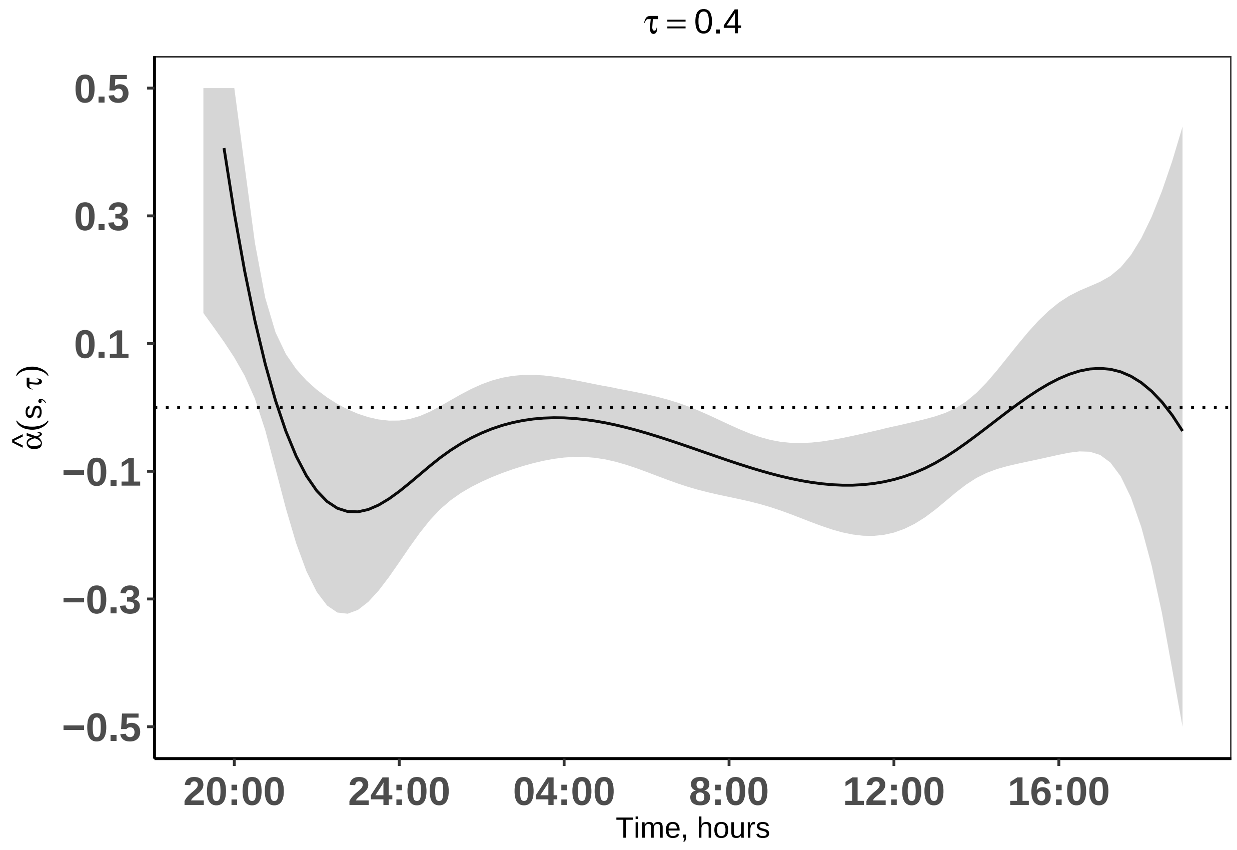

Figure 3: Estimated coefficient function $B_{rS}^T(s)\hat{\gamma}_S(\tau)$ with $d_S = 7$.