QTET QUINTET + 1

ROGER KOENKER

Abstract. A partially baked simulation experiment searching for a more informative design.

1. Introduction

To explore performance of the TMLE estimator of Diaz's "quantile treatment effect on the treated" (QTET) consider a simple simulation experiment. A binary treatment, T, is administered, there is a single, scalar, normally distributed covariate, X, and for each treated subject we observe a continuous response, Y_1, while for control subjects we see response, Y_0, where

$$Y_1 = \beta_0(U_1) + \beta_1(U_1)X + \gamma_0(U_1)T + \gamma_1(U_1)T \cdot X$$

and

$$Y_0 = \beta_0(U_0) + \beta_1(U_0)X$$

and (U_0, U_1) are marginally uniform random variables drawn from a one parameter (Frank) copula with dependence parameter, θ. Varying θ we can transit from independence, $\theta = 0$, to comonotonicity, $\theta \to \infty$. The conditional QTE is clearly

$$\mathrm{QTE}(\tau|X) = \gamma_0(\tau) + \gamma_1(\tau)X$$

and integrating out X over the treated subjects we have,

$$\mathrm{QTET}(\tau) = \gamma_0(\tau) + \gamma_1(\tau)\bar{X}_T$$

where $\bar{X}_T = \frac{1}{|\mathcal{T}|}\sum_{i\in\mathcal{T}} X_i$, and $\mathcal{T} = \{i : T_i = 1\}$.

In order to have any hope of estimating these quantities we must be willing to vouch for the fact that $T \perp\!\!\!\perp (Y_0, Y_1)|X$. In our experiment we consider two logistic specifications:

$$T = I(\alpha_0 + \alpha_1 \exp(X) \geq V),$$

and

$$T = I(\alpha_0 + \alpha_1 X \geq V),$$

with V logistically distributed. The conditional independence condition is satisfied in both cases, since T is assigned solely on the basis of the observable covariate, so no "compliance" issures are allowed to raise their ugly heads. However, the former case as implemented in `msim1.R` implies that the propensity score model used for the TMLE is misspecified.

Version: March 24, 2016.

When the pair (U_0, U_1) is not comonotonic the (dare I say?) classical[1] QR formulation of the treatment effect model

$$Q_Y(\tau|T, X) = \beta_0(\tau) + \beta_1(\tau)X + \gamma_0(\tau)T + \gamma_1(\tau)T \cdot X$$

is no longer valid, i.e., is no longer a proper quantile function, and we would like to understand better the consequences of this misspecification. Presumably, this could be investigated in a very partial way by varying the copula dependence parameter.

2. The Proof of the Pudding

Before launching a full comparison of our alternative estimators we first illustrate a graphical comparison of only the first two candidates. In Figure 1 we illustrate the true marginal QTET for our simulation DGP, along with two estimates: one based on the classical (aka ordinary) QR fitting of the model with treatment main and interaction effects, and the other based on fitting separate models for the treated and control subjects and then evaluating their difference at the mean covariate setting. Each *seems* to perform well, but as Hamlet says, ”Seems,” madam? Nay, it is; I know not ”seems.” Maybe, we should reserve judgement until there is more evidence.

```r
# A Simple Binary Treatment model
require(copula)
## Loading required package:  copula
require(quantreg)
DGP <- function(n, copula = frankCopula(1, dim = 2), IA = TRUE) {
    uv <- rCopula(n, copula)
    x <- exp(rnorm(n))
    X <- cbind(1,x)
    T <- (rlogis(n) < (X %*% c(0.5, -1)))
    y0 <- qnorm(uv[,1]) + x * (1 + qnorm(uv[,1])/3)
    if(IA)
       y1 <- qnorm(uv[,2]) + x * (1 + qnorm(uv[,2])/3) +
           T * (1 + log(x)/2) * qchisq(uv[,2],5)
    else
       y1 <- qnorm(uv[,2]) + x * (1 + qnorm(uv[,2])/3) +
           T * qchisq(uv[,2],5)
    y <- T * y1 + (1 - T) * y0
    data.frame(y = y, y0 = y0, y1 = y1, Treat = T, x = x)
}
tau <- 1:9/10
D <- DGP(1000)
f <- rq(y ~ log(x) * Treat, tau = 1:9/10, data = D)
f0 <- rq(y0 ~ log(x), tau = tau, data = D, subset = (Treat == 0))
```

[1] Here I employ “classical” in the familiar sense of “almost always wrong.”

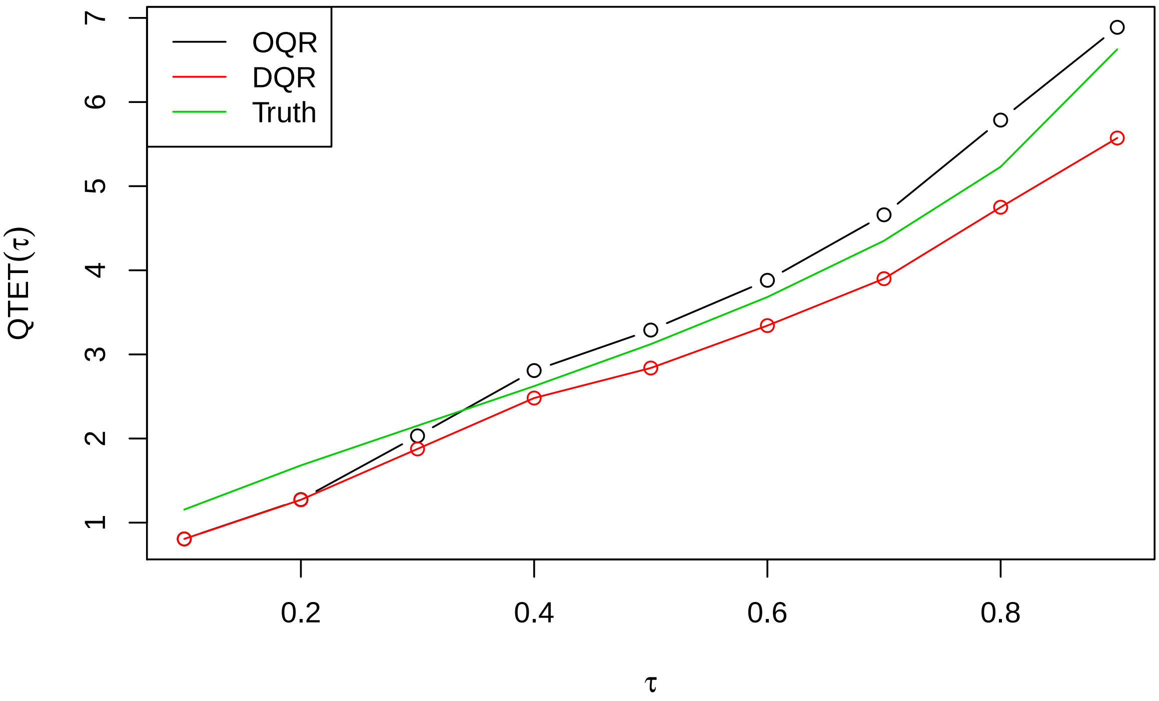

FIGURE 1. Marginal QTET's for Simulation Model 1

```r
f1 <- rq(y1 ~ log(x), tau = tau, data = D, subset = (Treat == 1))
qte1 <-  coef(f)[3,] + coef(f)[4] * mean(log(D$x[D$Treat == 1]))
plot(tau, qte1, type = "b", xlab = expression(tau),
     ylab = expression(QTET(tau)))
qte2 <-  t(coef(f1) - coef(f0)) %*% c(1, mean(log(D$x[D$Treat == 1])))
lines(tau, qte2, col = 2)
points(tau, qte2, col = 2)
lines(tau, qchisq(tau,5) * (1 + mean(log(D$x[D$Treat == 1])/2)), col = 3)
legend("topleft", c("OQR", "DQR","Truth"), col = 1:3, lty = 1)
```

Rather than keeping track of all the deciles as in Figure 1, we focus our next two experiments on the marginal median QTET. We consider four estimators in addition to the two described in Figure 1: the TMLE a "Naive" estimator, Firpo's onestep, and a simple onstep all described by Díaz (2015) and implemented by the `QTET` function presented in Koenker (2016). In Table 1 report mean bias and root mean squared error for our six estimators for the misspecified case that the treatment is assigned according to the

Table 1. Performance Comparison of Marginal QTETs for Simulation Model 1

	QR1	QR2	TMLE	Naive	Firpo	OneStep
Bias	0.483	0.0433	−0.547	−0.324	−0.366	−0.686
RMSE	0.597	0.2787	0.618	0.420	0.470	0.754

Table 2. Performance Comparison of Marginal QTETs for Simulation Model 2

	QR1	QR2	TMLE	Naive	Firpo	OneStep
Bias	0.572	0.310	−0.468	−0.170	−0.468	−0.657
RMSE	0.644	0.408	0.536	0.296	0.538	0.725

lognormal model, while the last four estimators all assume that the propensity score model should be linear $\log(X)$. In Table 2 this misspecification is corrected; treatment is assigned as assumed in the propensity score. This should improve performance of all the last four methods, but doesn't seem to do so as much as might be anticipated.

Appendix A. Simulation Protocol

The simulations underlying the results in Tables 1 and 2 were carried out in accordance with the protocol described in Koenker (2014) using the shell script `Rbatch`. Results for all 1000 replications of the experiment are saved, and tables and figures can then be easily prepared by loading these results. The code for these experiments and for making the tables is reproduced below.

```r
# A Simple Treatment model
require(copula)
require(quantreg)
source("QTET.R")
DGP <- function(n, copula = frankCopula(1, dim = 2), IA = TRUE) {
    uv <- rCopula(n, copula)
    x <- exp(rnorm(n))
    X <- cbind(1,x)
    T <- (rlogis(n) < (X %*% c(0.5, -1)))
    y0 <- qnorm(uv[,1]) + x * (1 + qnorm(uv[,1])/3)
    if(IA)
       y1 <- qnorm(uv[,2]) + x * (1 + qnorm(uv[,2])/3) +
           T * (1 + log(x)/2) * qchisq(uv[,2],5)
    else
       y1 <- qnorm(uv[,2]) + x * (1 + qnorm(uv[,2])/3) +
           T * qchisq(uv[,2],5)
    y <- T * y1 + (1 - T) * y0
    data.frame(y = y, y0 = y0, y1 = y1, Treat = T, x = x)
}
tau <- 1:9/10
R <- 1000
B <- matrix(0, 7, R)
for(i in 1:R){
    D <- DGP(1000)
    f <- rq(y ~ log(x) * Treat, tau = 1:9/10, data = D)
    f0 <- rq(y0 ~ log(x), tau = tau, data = D, subset = (Treat == 0))
    f1 <- rq(y1 ~ log(x), tau = tau, data = D, subset = (Treat == 1))
    qte1 <-  coef(f)[3,] + coef(f)[4] * mean(log(D$x[D$Treat == 1]))
    qte2 <-  t(coef(f1) - coef(f0)) %*% c(1, mean(log(D$x[D$Treat == 1])))
    truth <- qchisq(tau[5],5) * (1 + mean(log(D$x[D$Treat == 1])/2))
    Z <- data.frame(y = D$y, t = D$Treat, x = log(D$x))
    A <- QTET(Z)
    B[,i] <- c(truth, qte1[5], qte2[5], A$tmle, A$naive, A$firpo, A$onestep)
}
```

```
# Another Simple Treatment model
require(copula)
require(quantreg)
source("QTET.R")
DGP <- function(n, copula = frankCopula(1, dim = 2), IA = TRUE) {
    uv <- rCopula(n, copula)
    x <- exp(rnorm(n))
    X <- cbind(1,log(x))
    T <- (rlogis(n) < (X %*% c(0.5, -1)))
    y0 <- qnorm(uv[,1]) + x * (1 + qnorm(uv[,1])/3)
    if(IA)
       y1 <- qnorm(uv[,2]) + x * (1 + qnorm(uv[,2])/3) +
           T * (1 + log(x)/2) * qchisq(uv[,2],5)
    else
       y1 <- qnorm(uv[,2]) + x * (1 + qnorm(uv[,2])/3) +
           T * qchisq(uv[,2],5)
    y <- T * y1 + (1 - T) * y0
    data.frame(y = y, y0 = y0, y1 = y1, Treat = T, x = x)
}
tau <- 1:9/10
R <- 1000
B <- matrix(0, 7, R)
for(i in 1:R){
    D <- DGP(1000)
    f <- rq(y ~ log(x) * Treat, tau = 1:9/10, data = D)
    f0 <- rq(y0 ~ log(x), tau = tau, data = D, subset = (Treat == 0))
    f1 <- rq(y1 ~ log(x), tau = tau, data = D, subset = (Treat == 1))
    qte1 <-  coef(f)[3,] + coef(f)[4] * mean(log(D$x[D$Treat == 1]))
    qte2 <-  t(coef(f1) - coef(f0)) %*% c(1, mean(log(D$x[D$Treat == 1])))
    truth <- qchisq(tau[5],5) * (1 + mean(log(D$x[D$Treat == 1])/2))
    Z <- data.frame(y = D$y, t = D$Treat, x = log(D$x))
    A <- QTET(Z)
    B[,i] <- c(truth, qte1[5], qte2[5], A$tmle, A$naive, A$firpo, A$onestep)
}
```

```
load("msim1.Rda")
G <- B[-1,] - B[1,]
H <- rbind(apply(G,1,mean), sqrt(apply(G^2, 1, mean)))
dimnames(H) <- list(c("Bias", "RMSE"),
                    c("QR1", "QR2", "TMLE", "Naive", "Firpo", "OneStep"))
cap1 = "Performance Comparison of Marginal QTETs for Simulation Model 1"
latex(H, file = "", rowlabel = "", digits = 3, caption = cap1)
```

```
load("msim2.Rda")
G <- B[-1,] - B[1,]
H <- rbind(apply(G,1,mean), sqrt(apply(G^2, 1, mean)))
dimnames(H) <- list(c("Bias", "RMSE"),
                    c("QR1", "QR2", "TMLE", "Naive", "Firpo", "OneStep"))
cap2 = "Performance Comparison of Marginal QTETs for Simulation Model 2"
latex(H, file = "", rowlabel = "", digits = 3, caption = cap2)
```

References

Díaz I. 2015. Efficient estimation of quantiles in missing data models. *arXiv:1512.08110* .

Koenker R. 2014. A protocol for simulations in R. `http://www.econ.uiuc.edu/~roger/research/repro/protocol.pdf`.

Koenker R. 2016. Tmle: As i (dimly) understand it. Unpublishable.