CONFORMAL QUANTILE REGRESSION: AN R VINAIGRETTE

ROGER KOENKER

1. Introduction

Romano et al. (2019) have recently introduced a variant of conformal inference based on quantile regression. Their approach and implementation is based on machine earning methods, notably QR random forest and deep learning algorithms typically expressed in python. I thought it might be interesting to make an R implementation. I will focus on low dimensional settings that can be adaquately modeled with the `rqss` function from the **quantreg** package. The function `rqss` is a general purpose additive modeling function with univariate and bivariate functional components. Component functions are penalized by total variation of their derivatives.

The essential feature of the conformal inference approach is a sample splitting device that allows one to adjust a confidence band constructed with training data based on its performance on a validation sample. The R implementation using `rqss` is very simple and looks like this:

```
conformal <- function(formula, taus = c(0.05, 0.95), newdata, split = 0.5,
                        data = parent.frame(), method = "fn", ...){

    D <- data[,all.vars(formula)]
    s <- c(1,2,sample(1:nrow(D), floor(split * nrow(D))))
    qlo <- rqss(formula, tau = taus[1], data = D[s,], method = method, ...)
    qhi <- rqss(formula, tau = taus[2], data = D[s,], method = method, ...)
    x2 <- D[-s,]
    y2 <- D[-s,1]
    ylo2 <- predict(qlo, newdata = x2)
    yhi2 <- predict(qhi, newdata = x2)
    E <- pmax(ylo2 - y2, y2 - yhi2)
    QE <- quantile(E, diff(taus))
    pred <- NULL
    if(!missing(newdata))
        pred <- cbind(predict(qlo, newdata = newdata) - QE,
                    predict(qhi, newdata = newdata) + QE)
    list(qlo = qlo, qhi = qhi, QE = QE, pred = pred)
}
```

Estimates of lower and upper conditional quantile functions are made, predictions are then made for the validation sample observations, and finally a modified prediction interval is made for a potential newdata set. The main impediment to this seemed to be the fact that the current version of **quantreg** didn't have a `subset` argument for the `rqss` function. Fixing this involved relearning some tricks of the formula processing trade, but had the added benefit that it should have been there all along. In the end I realized that there was a simpler strategy that avoided using the `subset` argument, but c'est la vie.

Several other issues require some further investigation:

Version: September 22, 2020. A genre manifesto for R Vinaigrettes is available at `http://davoidofmeaning.blogspot.com/2016/12/r-vinaigrettes.html`.

- Another issue is that the predict method for `rqss` doesn't know how to extrapolate. I would be curious to know how other implementations deal with this, but I've not looked into it, yet. This issue is kludged in the example to follow by introducing two extreme elements of the training data.
- The function `rqss` relies on total variation penalized quantile regression with univariate terms in the additive model penalizing total variation of the first derivative of the estimated function. In the next example, taken from Romano et al. (2019) it will be evident that it might have been better to penalize the function itself rather than its derivative.
- The function `rqss` is intended to be a full service additive modeling function, however the current code hasn't been tested with multiple components, or for that matter for bivariate components.
- As usual it is difficult to select smoothing parameters for the TV penalization, whether machine earning methods can assist in this respect remains to be seen.
- The performance guarantee offered by the conformal predictions is based on average coverage over a new hypothetical sample like that of the training and validation samples, so it may provide little comfort to anyone looking for a narrower (or wider) focus on the design space.

```
# Code for Figure 1
# See: https://github.com/yromano/cqr/blob/master/cqr_synthetic_data_example_1.ipynb
require(quantreg)
set.seed(1729)
n = 7000
x = c(0,5, runif(n, 0, 5)) # Docs sez U[1,5], code sez U[0,5]
y = rpois(n+2, sin(x)^2 + 0.1) + 0.03*x*rnorm(n+2) + 25*(runif(n+2) < 0.01) * rnorm(n+2)
D = data.frame(y = y, x = x)
plot(x,y, cex = .25, ylim = c(-2,6), col = 'grey')
newx = seq(0,5,by = 0.05)
f = conformal(y ~ qss(x,lambda = .5), split = 2/7, data = D, newdata = list(x = newx))
plot(f$qlo, add = TRUE, col = 2, lwd = 3)
plot(f$qhi, add = TRUE, col = 2, lwd = 3)
lines(newx, f$pred[,1], col = 4)
lines(newx, f$pred[,2], col = 4)
```

It is interesting, perhaps, to compare the conformal band with more traditional confidence bands for the two conditional quantile functions illustrated in Figure 1. As described in Koenker (2011) it is relatively easy to constuct pointwise bands based on conditioning on the smoothing parameter λ, and it is only slightly more difficult to construct uniform bands using the Hotelling tube technique. It is striking to compare these bands, shown in Figure 2, with the conformal band and contemplate the differences in the promises they make about the arrival of new data.

```
# Code for Figure 2
plot(x,y, cex = .25, ylim = c(-2,6), col = 'grey')
plot(f$qlo, add = TRUE, col = 2, lwd = 3, bands = "both")
plot(f$qhi, add = TRUE, col = 2, lwd = 3, bands = "both")
```

2. Total Variation Penalization of g Itself

As noted above the RPC example begs for penalization of $TV(g)$ rather than $TV(g')$. This turns out to be remarkably easy to implement, so I've added an optional argument `Dorder` to the `qss` function. When `Dorder = 1` as in the default we penalize $TV(g')$ as illustrated above, while if `Dorder = 0`, $TV(g)$ is penalized instead. The resulting fit for the RPC example is illustrated in Figure 3, lending some support to the original conjecture.

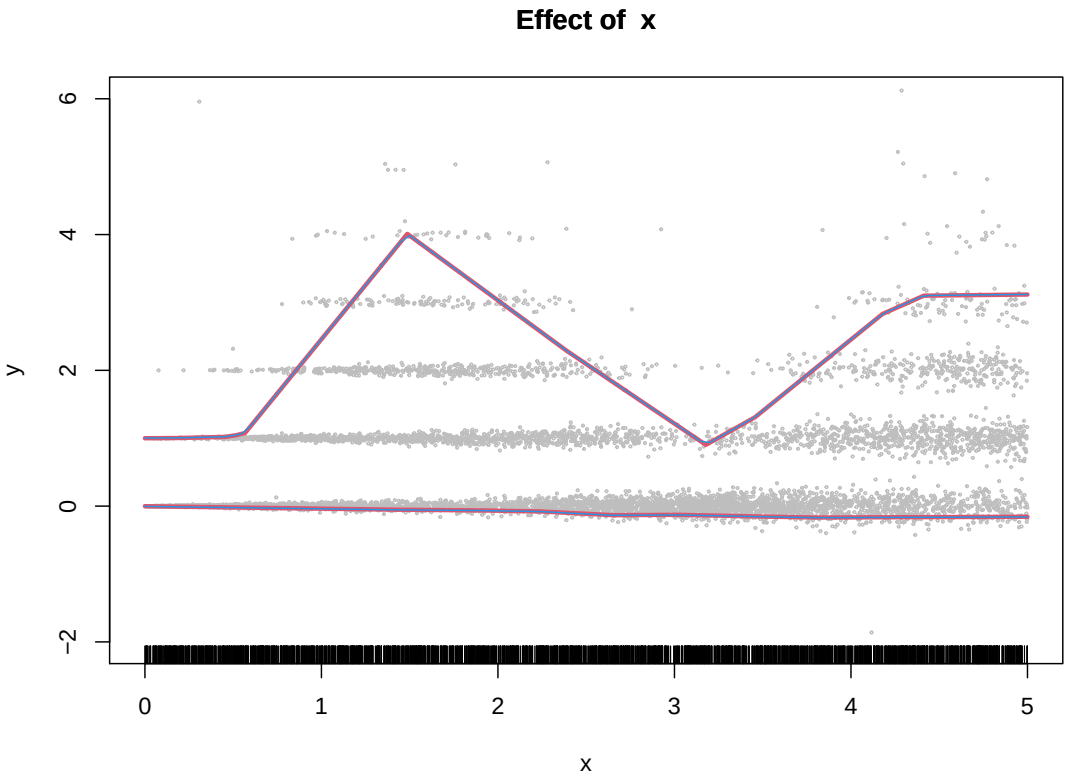

FIGURE 1. Example 1 of Romano, Patterson and Cand'es: As shown in the code below, the response is concentrated in bands determined by a Poisson component with some quite extreme outliers that are (mostly) invisible in this plot. The Poisson rate is periodic accounting for the obvious heteroscedasticity. The red curves depict the predicted 0.05 and 0.95 conditional quantile estimates based on the training data, while the blue curves depict the conformally modified estimates. In this example the conformity score E quite small and the conformal modification is negligible.

```
# Code for Figure 3
plot(x,y, cex = .25, ylim = c(-2,6), col = 'grey')
newx = seq(0,5,by = 0.05)
f = conformal(y ~ qss(x,lambda = .05, Dorder = 0), split = 2/7, data = D, newdata = list(x = newx))
plot(f$qlo, add = TRUE, col = 2, lwd = 3)
plot(f$qhi, add = TRUE, col = 2, lwd = 3)
lines(newx, f$pred[,1], col = 4)
lines(newx, f$pred[,2], col = 4)
```

3. CONCLUSION

Conformal inference for quantile regression is relatively easy to implement for the additive modeling formulation of the `rqss` function in R. However, there are several remaining issues, not the least of which is whether the performance guarantees offered by the method are relevant to practical decision making.

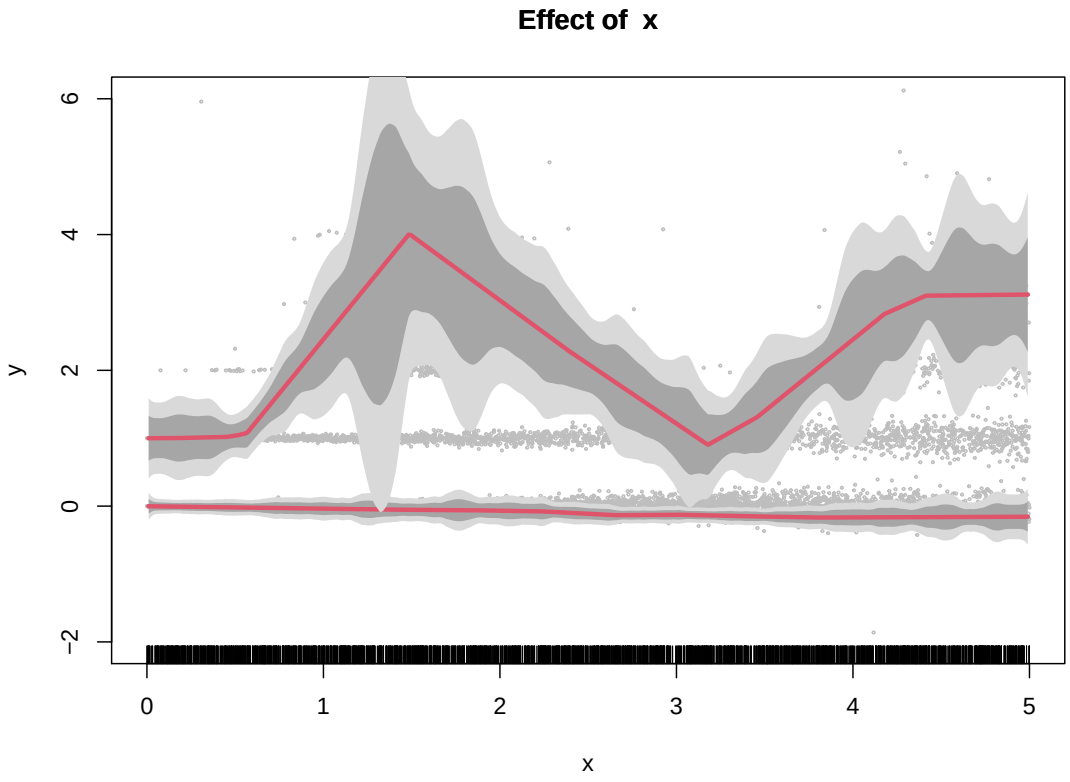

FIGURE 2. Pointwise and Uniform Confidence Bands for RPC Example: In contrast to the conformal prediction band, pointwise and uniform bands for the 0.05 and 0.95 conditional quantile functions are considerably wider. The uniform band is based on the Hotelling tube construction described in Koenker (2011) and is depicted as the light grey shaded band enclosing the darker grey pointwise band.

REFERENCES

Koenker, R. (2011), 'Additive models for quantile regression: Model selection and confidence bandaids', *Brazilian Journal of Probability and Statistics* **25**, 239–262.

Romano, Y., Patterson, E. and Candès, E. J. (2019), 'Conformalized quantile regression'. 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

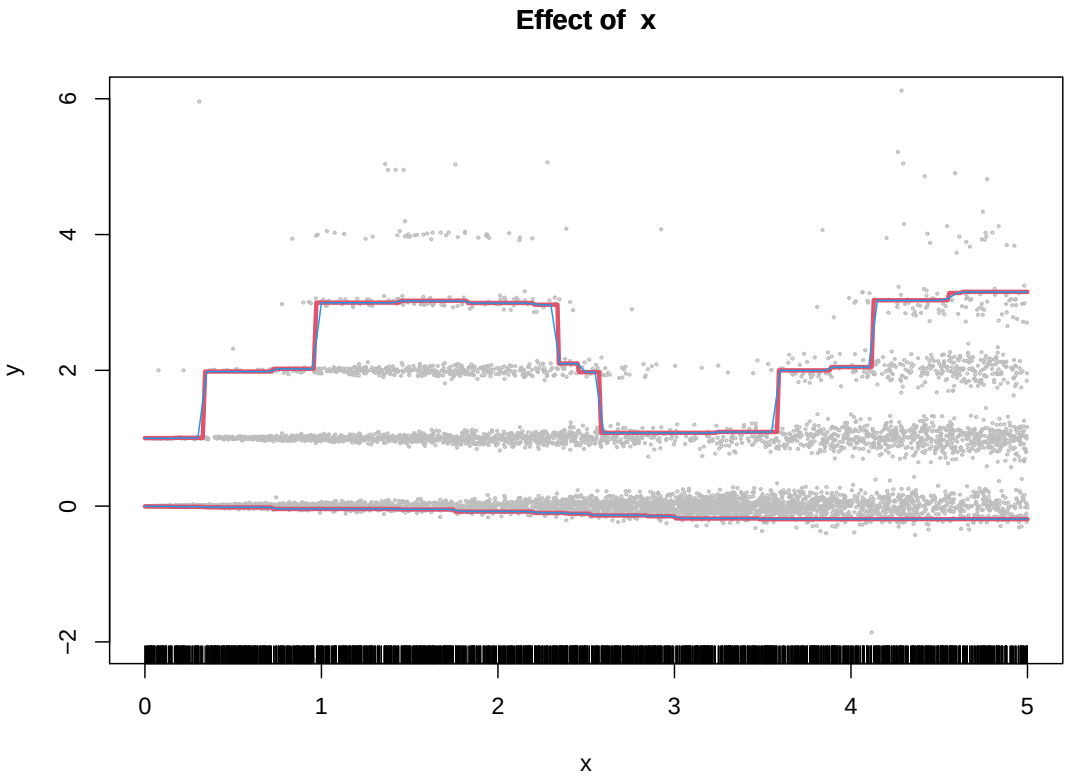

FIGURE 3. Example 1 of Romano, Patterson and Candès: As a contrast to the earlier piecewise linear fit obtained by total variation penalization of the first derivative of g. Here we plot the fit for a total variation penalized estimate of g itelf. Clearly this penalty is better suited to the example and mimics quite well the fit depicted in the RPC paper.