λ-SELECTION: AN R VINAIGRETTE

ROGER KOENKER

1. Introduction

Before I became obsessed with models for conditional quantile functions I was enamored by the work of Grace Wahba on smoothing splines. Eventually, there was an opportunity to combine these obsessions in Koenker et al. (1994) and Koenker and Mizera (2004) which replaced the classical L_2 smoothing penalty with an L_1 penalty that had the interpretation of penalizing the total variation of the first derivative of the fitted function or its gradient, respectively. These methods were implemented in my **quantreg** package and have been gradually developed over the years, the most important step in this development, described in Koenker (2011), was the introduction of methods for estimating and plotting confidence bands for the fitted functions. A glaring lacuna in this development, the fly in the ointment, so to speak, is that I never managed to provide a decent method for automatically selecting the parameters that control the degree of smoothness imposed by the methods.

It is true that in the papers referred to already, there was some mumbling about minimizing AIC/BIC like criteria, and hidden away in **quantreg** there is a `demo` that shows how to implement an AIC procedure for so-called λ selection for univariate components. This failure of personal courage was partially due to some early, traumatic experiences with GCV methods for classical smoothing splines. There are no shortage of proposals for λ selection in related problems, but it always seemed to be one of those problems that was best left to others.

Then, a few months ago, I received a note from someone at Netflix who mentioned that his colleagues had had some success using our TV smoothing methods in A/B testing for models of network latency. A bit of googling revealed a recent paper Zhang et al. (2020) that described this and some other successful applications of k-fold cross validation combined with the Kleiner et al. (2014) bag of little bootstraps idea for both univariate and bivariate smoothing. This provided an incentive to revisit the problem of λ selection in the hope that I could finally provide a viable implementation. I will briefly describe here some new experience with both the AIC approach and with a variant of k-fold cross validation. Unfortunately neither seems to offer the panacea that I was looking for.

2. The AIC Method

Nonparametric smoothing for conditional quantile functions is implemented in **quantreg** by the `rqss` function. This function is modeled after the `gam` function in the **mgcv** package

November 15, 2020. After circulating an initial draft of this note I received a very thoughtful email from Likun Zhamg the lead author of the Netflix paper who noted that in the very large samples that they were considering the MCV approach did tend to produce much smoother objective functions than those reported with much smaller samples in the present note. At the same time, the AIC procedures difficulties with identifying the "correct" dimensionality of the fitted model became more severe for larger sample sizes. Both of these points seem quite persuasive to me and I hope to pursue the MCV selection approach in future work.

of Wood (2017). The model is specified by a formula that describes a series of additive terms: nonparametric terms are represented by contributions that look like `qss(x, lambda = lam)`, where `x` is a conditioning covariate, and `lambda` is the smoothing parameter for the penalty contribution,

$$\lambda P(g) = \lambda TV(g') = \lambda \int |g''(x)| dx,$$

and linear (parametric) terms appear naked separated by "+" signs. (Inside the `qss` terms one can also restrict the fitted g functions to be monotone, or convex, or concave, but I will not dwell further on this, here.) The idea behind the AIC method of selecting λ's is to define an effective dimensionality of the fitted model and use that to construct an objective function to be minimized, as λ increases the fitted g is forced to become more linear. Effective dimensionality of any $\hat{g}$ can be defined in several ways. Penalizing total variation of g' implies that the $\hat{g}$ must be piecewise linear, so we can simply count the number of distinct pieces for each additive, nonparametric term. Alternatively, one can simply count the number of interpolated observations to compute the effective dimensionality of the model as a whole. The adverb "simply" is used ironically here in view of the fact that both of the aforementioned procedures rely on dubious features of finite precision arithmetic. Distinguishing what is zero from what is small is notoriously difficult. It should be noted however that this is no different from the situation often recommended for L_2 penalized smoothers where the trace of the quasi-projection matrix plays this role. An elegant unification of such estimates of dimensionality is provided by the proposal of Meyer and Woodroofe (2000) to use

$$\dim(\hat{g}) = \mathrm{div}(\hat{g}) = \sum_{i=1}^{n} \partial \hat{g}(x_i) / \partial y_i.$$

In the case of quantile regression this formula reduces to "simply" counting the number of interpolated points, since $\hat{g}$ is insensitiive, locally, to movements of the uninterpolated points.

Given an estimate, p_λ, of the dimensionality of the fitted model, the AIC criterion may be expressed as,

$$AIC(\hat{g}_\lambda, k) = 2n \log(n^{-1} \sum_{i=1}^{n} \rho_\tau(y_i - \hat{g}_\lambda(x_i))) + kp_\lambda.$$

The original AIC criterion would use $k = 2$, while the Schwarz criterion would use $k = \log n$.[1] When `k < 0` the Schwarz option is employed. In the example illustrated in Figure 1, I've followed my past inclination and used the Schwarz variant of the AIC criterion that tends to produce somewhat smoother estimates.

In Figure 1 I've illustrated an example of this sort of AIC fitting. As can be seen in the R code, the data illustrated in the right panel has been generated by a simple location-scale model with logarithmic mean and Gaussian noise. The optimization of AIC is carried out with the R function `optimize`, but as can be seen in the left panel where AIC(λ) is plotted on an equally spaced grid the objective is quite rough. This makes the problem of optimization quite challenging: an optimistic view would be that the fit isn't very sensitive to the choice of λ in the region from about 0.5 to 1.5, so the utility of a highly accurate optimization can

[1] As usual we are playing fast and loose with additive and multiplicative constants that are independent of λ here, see Koenker (2016) for gory details. Since we are just looking for where the criterion is minimized such constants are irrelevant.

easily be exaggerated. On the other hand if one imagines solving for several λ's in a more complicated additive model the problem seems more daunting.

```
arqss <- function(x, y, tau, interval){
    g <- function(lam, y, x, tau)
        AIC(rqss(y ~ qss(x, lambda = lam),tau = tau),k = -1)
    lamstar <- optimize(g, interval, x = x, y = y, tau = tau)
    rqss(y ~ qss(x, lambda = lamstar$min))
}
n <- 200
x <- sort(runif(n, 0, 20))
g0 <- function(x, tau)
    log(x) + 0.2*(log(x))^3 + log(x) * qnorm(tau)/4
y <- g0(x, runif(n))
par(mfrow = c(1,2))
lams <- aics <- seq(0.025, 10, by = 0.2)
for(i in 1:length(lams))
    aics[i] <- AIC(rqss(y ~ qss(x,lambda = lams[i])), k = -1)
plot(lams, aics, cex = .5, lwd = 2, type = 'l',
    xlab = expression(lambda), ylab = expression(AIC( lambda )))
plot(x, y, cex = .5, col = "grey")
f <- arqss(x, y, 0.5, c(0.01, 10))
plot(f, add = TRUE, lwd = 2)
lines(x,g0(x, 0.5),col = "red", lwd = 2)
lam <- round(f$lambda,3)
text(10, 1, bquote(lambda == ~  .(lam)))
```

3. The k-fold Cross Validation Method

The idea of multi-fold cross validation (MCV) was introduced by Geisser (1975) as an alternative to the classical jackknife, and has been widely adopted for many nonparametric smoothing parameter selection problems. Zhang et al. (2020) argue convincingly that it can be usefully adapted to large-scale nonparametric quantile regression settings as well. The motivation is quite compelling: rather than the traditional delete-1 jackknife, which has well-known efficiency problems in situations where the Sherman, Morrison Woodbury formula isn't available, deleting larger groups can be implemented quite efficiently in a much larger class of problems, especially so since it is quite easily parallelizable. I was quite curious to explore this further.

A basic implementation is quite easy. The parameter k is used to divide the observations into k groups, $V_1, \ldots, V_k$ each of approximate size n/k. The model is fit by omitting each of these groups, in turn, and computing a measure of prediction error for the omitted group. The predict method for `rqss` doesn't understand how to extrapolate so any observations that require extrapolation have to be removed before the prediction error is computed. The prediction errors are then aggregated into a final prediction error for each choice of λ and then minimized over λ. This procedure is illustrated in Figure 2 using the same test problem used for the AIC criterion. As we saw for AIC the MCV criterion also produces a very rough objective function, however since we are minimizing over the gridded values of λ in this case,

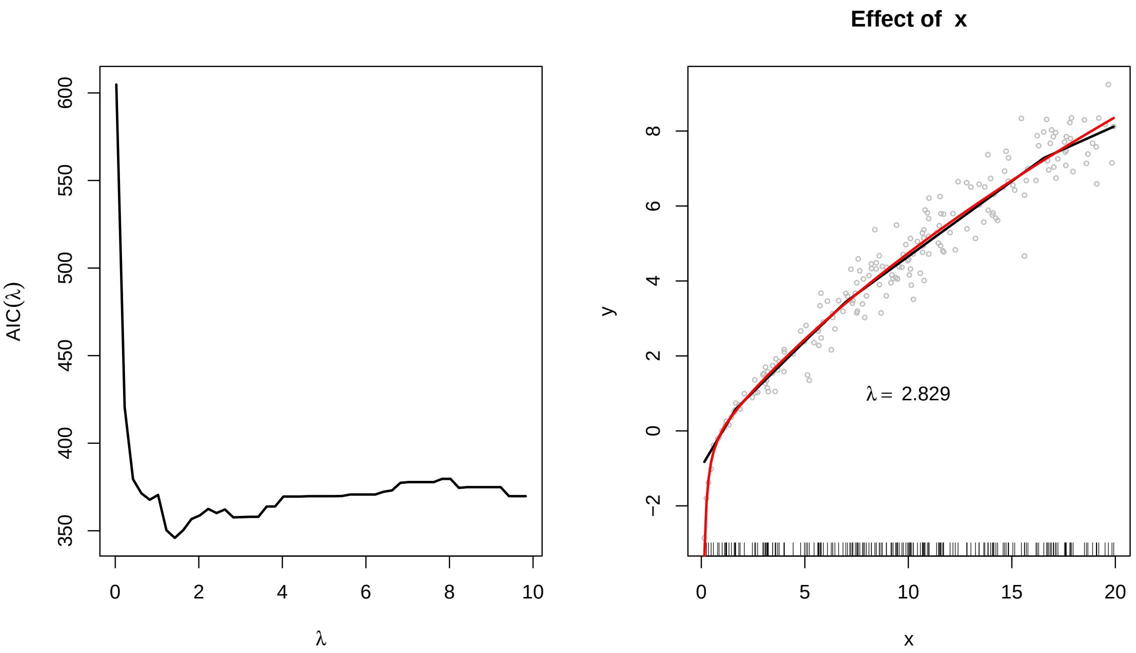

Figure 1. Automatic λ Selection by AIC Criterion: Left panel plots the AIC criterion as a function of λ, while the right panel plots the estimated $\hat{g}_\lambda$ for the selected λ^*. Optimize clearly doesn't find a global optimum here.

we happen to select a much smaller λ than with AIC. It would be hard to argue that it is a better λ, however. It yields a somewhat rougher fitted function.

```
MCV <- function(lambdas, formula, data, tau = 0.5, k = 10){
    F <- Munge(formula, lambdas = lambdas)
    f <- rqss(F, data, tau = tau)
    n <- f$n
    m <- length(f$qss)
    y <- f$y[1:n]
    folds = sample(rep(1:k, length = n))
    U = NULL
    for(i in 1:k){
        s = which(folds != i)
        M = rqss(F, data = data[s,], tau = tau)
        nd = data[-s,]
        G = matrix(0,nrow(nd),m)
        for(j in 1:m){ #remove extrapolates, if any
            g = f$qss[[j]]$xyz[,1]
            G[,j] = (min(g[s]) < g[-s]) & (g[-s] < max(g[s]))
        }
        h = as.logical(apply(G,1,prod))
        u = predict(M, newdata = nd[h,]) - (y[-s])[h]
        U = c(U,(u * (tau - (u < 0))))
    }
    mean(U)
}
n <- 200
x <- sort(runif(n, 0, 20))
g0 <- function(x, tau)
    log(x) + 0.2*(log(x))^3 + log(x) * qnorm(tau)/4
y <- g0(x, runif(n))
D <- data.frame(y = y, x = x)
lams <- mcvs <- seq(.02, 10, by = 0.2)
for(i in 1:length(mcvs))
    mcvs[i] <- MCV(lams[i], y ~ qss(x, lambda = lambdas[1]), D)
par(mfrow = c(1,2))
plot(lams, mcvs, cex = .5, lwd = 2, type = 'l',
    xlab = expression(lambda), ylab = expression(MCV( lambda )))
lambdastar <- lams[which.min(mcvs)]

plot(x, y, cex = .5, col = "grey")
f <- rqss(y ~ qss(x, lambda = lambdastar), data = D)
plot(f, add = TRUE, lwd = 2)
lines(x,g0(x, 0.5), col = "red", lwd = 2)
text(10, 1,bquote(lambda == ~  .(lambdastar)))
```

4. Conclusion

A theme of these Vinaigrettes is that numerical results like those we have illustrated here prove nothing, by themselves. We could repeat the exercise with larger sample sizes, different test functions or different distributional assumptions, but the unfortunate fact remains that it is difficult to choose λ's. The more there are, the more difficult it will be. Perhaps we can console ourselves with the countervailing fact that it doesn't matter much, even half-hearted optimization of a rather nasty objective function can produce a decent answer.

References

Geisser, S. (1975), 'The predictive sample reuse method with applications', *J. of Am. Statistical Association* **70**, 320–328.

Kleiner, A., Talwalkar, A., Sarkar, P. and Jordan, M. I. (2014), 'A scalable bootstrap for massive data', *Journal of the Royal Statistical Society: Series B* **76**, 795–816.

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

Koenker, R. (2016), 'Model selection and fishing for significance'. Available from `http://www.econ.uiuc.edu/~roger/courses/508/lectures/L4.pdf`.

Koenker, R. and Mizera, I. (2004), 'Penalized triograms: Total variation regularization for bivariate smoothing', *Journal of the Royal Statistical Society: Series B* **66**, 145–163.

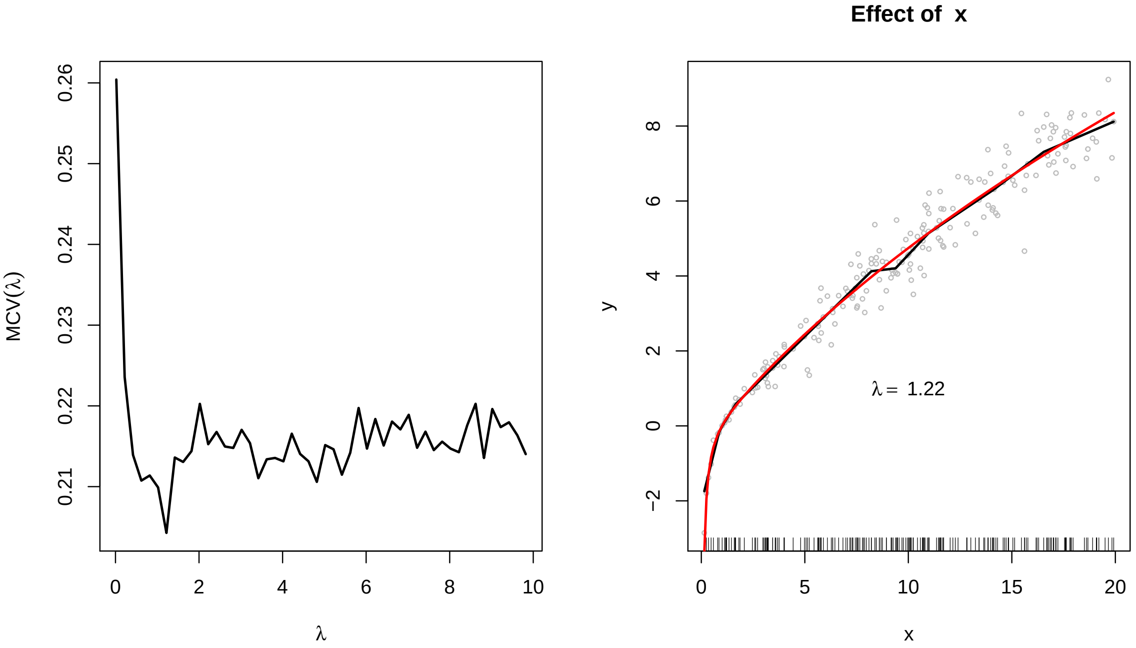

Figure 2. Automatic λ Selection by MCV Criterion: Left panel plots the MCVC criterion as a function of λ, while the right panel plots the estimated $\hat{g}_\lambda$ for the selected λ^*

Koenker, R., Ng, P. and Portnoy, S. (1994), 'Quantile smoothing splines', *Biometrika* **81**, 673–80.

Meyer, M. and Woodroofe, M. (2000), 'On the degrees of freedom in shape-restricted regression', *Annals of Statistics* **28**, 1083–1104.

Wood, S. (2017), *Generalized Additive Models: An Introduction with R*, 2 edn, Chapman and Hall/CRC.

Zhang, L., Castillo, E. D., Berglund, A. and Tingley, M. (2020), 'Computing confidence intervals from massive data via penalized quantile smoothing splines', *Computational Statistics and Data Analysis* **144**, 106885.