/

COMPUTING SAMPLE QUANTILES: AN R VINAIGRETTE

ROGER KOENKER

1. Introduction

A classical problem in computational statistics is: how to compute the τth sample quantile? It seems ridiculously simple: don't you just sort the observations and find the $k = \tau n$ smallest one. Maybe, but not if you are interviewing for a job at Google. Sorting is quite efficient, and can be done in $O(n \log n)$ comparisons, but why would we want to fully sort the observations when we only need to care about a small range of them?

Linear methods began appearing in the 1960s, notably Hoare (1961), an adaptation of the better known "quicksort," Hoare (1962). An elegant analysis of Eppstein (2007) shows that "quickselect" requires an expected number of comparisons,

$$[2n(1 + \log(n/(n-k)) + 2k \log((n-k)/k))(1 + o(n)),$$

or, in the median, $k = n/2$, worst case, $2n(1 + \log(2) + o(1)) \leq 3.3863n + o(n)$ comparisons.

Floyd and Rivest (1975) proposed an algorithm[1] that improved upon Hoare's "quickselect," and provided an implementation in Algol 68. Subsequent work by Kiwiel (2005) established an expected performance bound of $n + \min(k, n-k) + O(\sqrt{n})$ for this algorithm. In 1996 I made a very literal translation of the Floyd-Rivest "select" algorithm from Algol to Ratfor, a preprocessor for Fortran, and incorporated it into my Splus package for quantile regression. This implementation involved Fortran that made a recursive call (to itself). This was formally a no-no in Fortran 77, but for a time certain compilers tolerated it. After the transition to R about 2001, at some point compilers on some CRAN test machines began to baulk at this recursive feature and I had to remove it from the package. Coincidently, around this time I had an email exchange with Krzysztof Kiwiel who had been working on "select" and had run across my Fortran version. His paper, Kiwiel (2005) clarified several aspects of the theoretical performance of the "select" algorithm and also provided a Fortran version that circumvented the recursive call. He kindly gave me permission in 2006 to include his code as part of my **quantreg** package. In a somewhat quixotic effort to convince R-core to replace the extant `quantile` function, I wrote interface code to produce all nine varieties of quantiles as described in the R man page for `quantile`. It was eventually judged to be insufficiently faster to justify the switch.[2]

In Koenker (2020) I recently tried to review the ecosystem of methods in the **quantreg** package for computing quantile regression estimates and evaluating their precision. This rekindled my interest in computation of univariate quantiles, and in particular the weighted

January 13, 2021. A genre manifesto for R Vinaigrettes is available at http://davoidofmeaning.blogspot.com/2016/12/r-vinaigrettes.html.

[1] There seems to be some confusion in the literature about the terminology of these early contributions. Hoare's "quickselect" is sometimes referrred to as "quickfind," and an early paper of Tibshirani (2008) misattributes "quickselect" to Floyd and Rivest. Knuth (1998) offers an authoritative account.

[2] It always amazed me that there were nine such varieties to resolve what was, after all, an ambiguity of measure zero. Empson (1930) could only come up with seven.

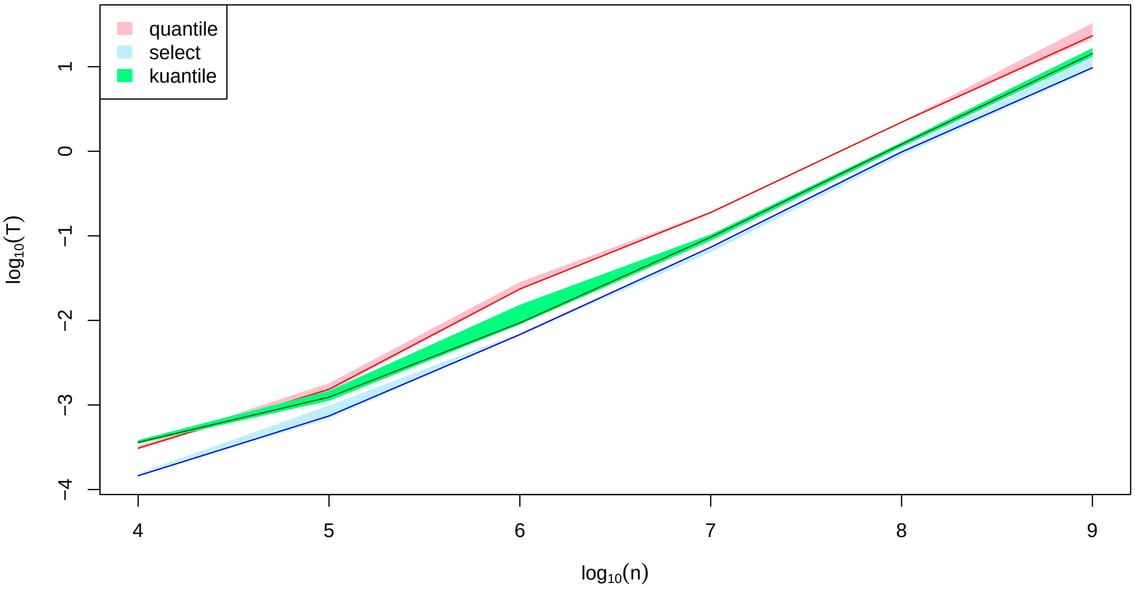

Figure 1. CPU Time (in seconds) for Computing a Single Quantile as a Function of Sample Size

quantiles that correspond to solving univariate quantile regression problems that force the fitted quantile regression line to be a ray through the origin. In the remainder of this note I'd like to first review the the classical univariate sample quantile problem, and then briefly turn to the weighted quantile problem.

2. Univariate Sample Quantiles

My interface to the Kiwiel code, because in part it accounts for all nine varieties of quantiles, seemed a bit bulky, so I wondered whether something closer to the original implementation of Floyd and Rivest might perform somewhat better. In the interim I had learned that Fortran 90 had reinstated the possibility of recursive calls. So I thought I would experiment with my old literal translation as a warm-up excercise before embarking on the weighted quantile coding that was my ultimate aim. The modification required for this was essentially just to add the word "recursive" before subroutine, The directory `src/ratfor` in the package **quantreg** contains the new version of the code in Ratfor.

In Figure 1 I compare the cpu time required to compute a single quantile for samples of various sizes with the functions, `quantile` from base-R, `kuantile` from the **quantreg** package, and `q489` from the **quantreg** package. The shaded regions represent the band between the first and third quantiles for each method based upon 50 replications and measured by the `microbenchmark` function. Both **quantreg** procedures exhibit a clear advantage over the base-R method with `q489` having a slight advantage due perhaps to the overhead entailed by the more elaborate R interface of the `kuantile` method.

3. Weighted Quantiles

In their simplest form weighted quantiles arise when we have repeated values of some observations and would like to use case weights to account for this repetition. More generally, we may wish to consider more general weights for each observation. These weights should be non-negative to preserve the convexity of the underlying optimization problem. One interpretation of the general weighted quantile estimate is to consider the problem,

$$\min_{a \in \mathbb{R}} \sum_{i=1}^{n} \rho_\tau(y_i - x_i a)$$

this is sometimes referred to as "regression through the origin" since we seek a scalar slope coefficient to minimize the usual quantile regression objective function. In effect this is a weighted quantile problem with observations $z_i = y_i/x_i$ and weights $w_i = |x_i|$. To solve such problems we need to order the z_i's and find the element such that the cumulative sum of the ordered weights exceeds a theshhold, a concise implementation of this strategy is given in the following R code.

```
wquantile <- function(x, y, t = 0.5) {
        ord <- order(y/x)
        z <- (y/x)[ord]
        wabs <- abs(x[ord])
        k <- sum(cumsum(wabs) < ((t - 0.5) * sum(x) + 0.5 * sum(wabs)))
        z[k + 1]
}
```

Of course, this requires a full sort of the z_i's so there should be a more efficient algorithm that operates like the Floyd-Rivest "select" method. A UK astrophysics project called "Starlink" produced such an open source implementation called `KPG1_QNTLx`, unfortunately my attempts to link it via R to do comparisons floundered; for sample sizes up to about 1,000,000 it performs quite well, but for larger sample sizess it segfaulted with a "memory not mapped" error. I was not able to track down the source of this error despite attempts to explore with my limited knowledge of fortran debugging with gdb and valgrind. As an alternative I wrote a version of "select" employing weights that was almost successful; it was relatively quick, didn't segfault at large sample sizes but unfortunately didn't quite get the right answer. It was typically only wrong by one or two order statistics, but again I lost patience trying to determine what was going wrong. Intrepid readers, if any, interested in pursuing this are encouraged to take a closer look at the code which will be made available with the pdf of this document.

4. Updating Sample Quantiles

Given an sample quantile based on n observations, we might also ask whether there is an efficient way to update our estimate when presented with new observations. This was the first example in the celebrated Robbins and Monro (1951) paper on stochastic approximation. There are several variants of this problem. One approach is closely tied to the Robbins-Monro algorithm and has been investigated by Holst (1985), Toulis et al. (2020) and others. Space efficient estimation methods are proposed by Tierney (1983), and Rousseeuw and Bassett (1990). Estimation of quantiles in distributed networks has been explored by Chambers et al. (2006), and Hammer et al. (2019), among others.

5. Conclusion

Quantiles play a crucial role in many statistical applications and larger sample sizes inevitably pose new challenges. My own interest in efficient computation of weighted sample quantiles stemmed from an old realization that the bounded variable simplex method of Barrodale and Roberts (1974) for solving ℓ_1 regression problems, and *a fortiori* other quantile regression problems, is essentially gradient descent combined with a step length optimization solving a weighted quantile problem. See Koenker (1996) for further details.

References

Barrodale, I. and Roberts, F. (1974), 'Solution of an overdetermined system of equations in the ℓ_1 norm', *Communications of the ACM* **17**, 319–320.

Chambers, J. M., James, D. A., Lambert, D. and Vander Wiel, S. (2006), 'Monitoring networked applications with incremental quantile estimation', *Statistical Science* **21**, 463–475.

Empson, W. (1930), *Seven Types of Ambiguity*, Chatto and Windus.

Eppstein, D. (2007), Blum-style analysis of quickselect. Available from `https://11011110.github.io/blog/2007/10/09/blum-style-analysis-of.html`.

Floyd, R. W. and Rivest, R. L. (1975), 'Algorithm 489: The algorithm select for finding the ith smallest of n elements', *Comm. ACM* **18**, 173.

Hammer, H. L., Yazidi, A. and Rue, H. (2019), 'A new quantile tracking algorithm using a generalized exponentially weighted average of observations', *Applied Intelligence* **49**.

Hoare, C. A. R. (1961), 'Algorithm 65: Find', *Comm. ACM* **4**, 321–322.

Hoare, C. A. R. (1962), 'Quicksort', *Commuter Journal* **5**, 10–16.

Holst, U. (1985), Recursive estimation of quantiles, *in* G. Lindgren, ed., 'Contributions to Probability and Statistics in Honour of Gunar Blom', Lund University Press, pp. 179–188.

Kiwiel, K. C. (2005), 'On floyd and rivest's select algorithm', *Theoretical Computer Science* **347**, 214–238.

Knuth, D. K. (1998), *The Art of Computing Programming: Sorting and Searching*, Vol. 3, 2nd edn, Addison-Wesley.

Koenker, R. (1996), rqx: Barrodale and Roberts lite. Available from: `http://www.econ.uiuc.edu/~roger/research/rq/rqx.R`.

Koenker, R. (2020), Quantile regression methods: An R vinaigrette. Available from `http://www.econ.uiuc.edu/~roger/research/vinaigrettes/vinaigrette.html`.

Robbins, H. and Monro, S. (1951), 'A stochastic approximation method', *Annals of Mathematical Statistics* **22**, 400–407.

Rousseeuw, P. J. and Bassett, G. W. (1990), 'The remedian: A robust averaging method for large data sets', *Journal of the American Statistical Association* **85**, 97–104.

Tibshirani, R. J. (2008), Fast computation of the median by successive binning. available from: urlhttps://www.stat.cmu.edu/ ryantibs/papers/median.pdf.

Tierney, L. (1983), 'A space-efficient recursive procedure for estimating a quantile of an unknown distribution', *SIAM Journal on Scientific and Statistical Computing* **4**, 706–711.

Toulis, P., Horel, T. and Airoldi, E. M. (2020), 'The proximal Robbins–Monro method', *Journal of the Royal Statistical Society: Series B* .