

# Hints on lmcholsolve

yifan

May 5, 2014

lmcholsolve in optimise2 will solve the following question using Cholesky-decomposition:

$$H\beta = y \quad H > 0 \quad (1)$$

It is a little bit tricky to compare the performances of `base::solve` and `optimise2::lmcholsolve`. The original `solve` used in R relies on the BLAS you choose. On my Mac machine, I use vecBLAS and openBLAS with MacPro1,1(2 X Intel(R) Xeon(R) CPU X5355 2.66GHz), the performances are shown in Fig[?]. Although `solve` uses all 8 cores <sup>1</sup>, `optimise2::lmcholsolve` is

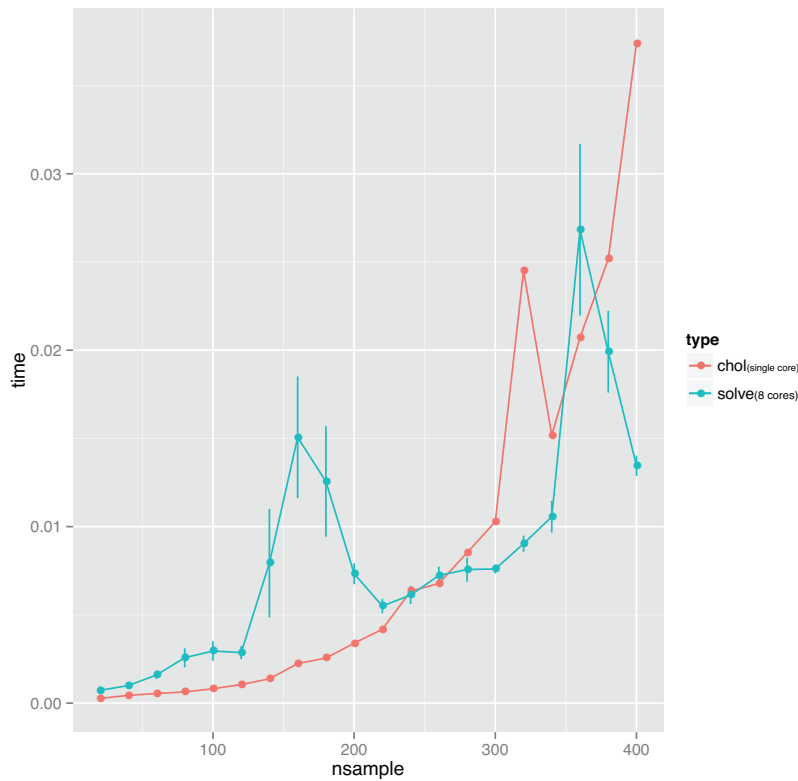


Figure 1: 100 simulation.

still not so bad when sample size is less than 300. Parallel QR-decomposition/householder transformation may boost the solution, but it is not of my concern now. One thing I must state is the function will not check the following conditions are true or not:

- 1  $H \geq 0$  (or even it is a square matrix);
- 2 the dimensions of  $H$  and  $y$  agree with each other.

In **serious** studies, one may use *svd* instead of sweeping/householder/QR as *svd* tends to do a better job than others in the sense of accuracy and stability. The weakness of *svd* is speed. Generally speaking, if the design matrix is nearly singular, sweeping, which is used in SAS, performs the worst among all.

---

<sup>1</sup>In fact , it should be 4 cores 8 threads