

Conditional autoregressive model: a comparison with `hglm`

The vignette of the `hglm` package (Rönnegård et al., 2010), version 2.2.1, presents comparisons of fitting times between `hglm` and `spaMM`, using an unspecified version of `spaMM`. `spaMM` has actually long performed much better than reported there, particularly when using the now-recommended function `fitme`. Here are cumulative timings over 10 samples for each of 6 different designs (CAR models over 50 to 500 nodes as in the original comparison; the source code is found below. These comparisons were made with `hglm` version 2.2.1 and `spaMM` version 3.9.12 using a past version of R, and would be awkward to update because `hglm` appears to fail to fit any of the samples on more recent development versions of R.

##	time_hglm	time_HLCor	time_fitme
## n50	NA	0.45	0.81
## n100	0.84	0.67	1.31
## n200	5.17	2.51	1.73
## n300	21.17	10.00	1.73
## n400	43.99	22.80	2.09
## n500	87.88	34.81	2.56

The initial NA reflects the fact that `hglm` fails on one of the samples (a fact which is hidden in the original report). `spaMM::HLCor` is generally faster than `hglm`; and `spaMM::fitme` is faster for the largest data sets, for reasons given in Section 3.5.3 of the “gentle introduction” to `spaMM`. More elaborate comparisons could also highlight the stricter numerical criteria used by `spaMM`.

References

Rönnegård, L., Shen, X., and Alam, M. 2010. `hglm`: A package for fitting hierarchical generalized linear models, *R Journal* **2**, 20–27.

```

## require(hglm)
## require(spaMM)
## spaMM.options(sparse_precision = NULL)
##
## data(ohio)
## nrepsim <- 10
## thglm <- tfitme <- tHLCor <- data.frame(n50 = rep(NA, nrepsim),
##   n100 = rep(NA, nrepsim), n200 = rep(NA, nrepsim), n300 = rep(NA,
##     nrepsim), n400 = rep(NA, nrepsim), n500 = rep(NA, nrepsim))
## n <- c(50, seq(100, 500, 100))
## rngcheck <- ("sample.kind" %in% names(formals(RNGkind)))
## if (rngcheck) suppressWarnings(RNGkind("Mersenne-Twister", "Inversion",
##   "Rounding"))
## for (k in 1:6) {
##   set.seed(911)
##   for (i in 1:nrepsim) {
##     lv <- sample(levels(ohioMedian$district), n[k])
##     idx <- which(ohioMedian$district %in% lv)
##     subdistrict <- factor(as.character(ohioMedian$district)[idx])
##     subMedian <- data.frame(MedianScore = ohioMedian$MedianScore[idx],
##       district = subdistrict)
##     subD <- ohioDistrictDistMat[levels(subdistrict), levels(subdistrict)]
##     wrapped <- try(system.time(hg1 <- hglm(fixed = MedianScore ~
##       1, random = ~1 | district, rand.family = CAR(D = subD),
##       data = subMedian, method = "EQL1"))))
##     if (!inherits(wrapped, "try-error"))
##       thglm[i, k] <- wrapped[3]
##     tfitme[i, k] <- system.time(sp1 <- fitme(MedianScore ~
##       1 + adjacency(1 | district), data = subMedian, method = "HL(1,0)",
##       adjMatrix = subD))[3]
##     tHLCor[i, k] <- system.time(sp1 <- HLCor(MedianScore ~
##       1 + adjacency(1 | district), data = subMedian, HLmethod = "HL(1,0)",
##       adjMatrix = subD))[3]
##     cat(i, " ")
##   }
## }
## if (rngcheck) RNGkind("Mersenne-Twister", "Inversion", "Rejection")
##
## (CAR_timings <- structure(data.frame(time_hglm = colSums(thglm),
##   time_HLCor = colSums(tHLCor), time_fitme = colSums(tfitme)),
##   hglm_version = packageVersion("hglm"), spaMM_version = packageVersion("spaMM")))

```