

speedtest

Benjamin Christoffersen

2018-06-22

Setup

```
library(dynamichazard); library(microbenchmark)

## Loading required package: survival

sim_func <- function(n, p){
  func <- asNamespace("dynamichazard")$test_sim_func_logit
  set.seed(101)
  t_max <- 30L
  func(n_series = n, n_vars = p, t_max = t_max, x_range = 1, x_mean = 0,
       beta_start = runif(p, -1.5, 1.5),
       intercept_start = -3, sds = c(.1, rep(.25, p)),
       tstart_sampl_func = function(t0, t_max)
         max(0, runif(1, -t_max, t_max - 1L)),
       lambda = 1 / 10)
}

get_rune_time_summary <- function(n, p){
  sims <- sim_func(n, p)

  out <- summary(microbenchmark(
    EKF_one_correction_step =
      suppressMessages(ddhazard(
        formula = Surv(tstart, tstop, event) ~ . - id,
        data = sims$res,
        model = "logit",
        id = sims$res$id,
        by = 1L,
        max_T = 30L,
        Q_0 = diag(1e6, p + 1L),
        Q = diag(1e-1, p + 1L))),
    EKF_more_correction_step =
      suppressMessages(ddhazard(
        formula = Surv(tstart, tstop, event) ~ . - id,
        data = sims$res,
        model = "logit",
        id = sims$res$id,
        by = 1L,
        max_T = 30L,
        Q_0 = diag(1, p + 1L),
        Q = diag(1e-1, p + 1L),
        control = list(NR_eps = 1e-3))),
    SMA = suppressMessages(ddhazard(
```

```

    formula = Surv(tstart, tstop, event) ~ . - id,
    data = sims$res,
    model = "logit",
    id = sims$res$id,
    by = 1L,
    max_T = 30L,
    Q_0 = diag(1e6, p + 1L),
    Q = diag(1e-1, p + 1L),
    control = list(method = "SMA"))),

GMA = suppressMessages(ddhazard(
  formula = Surv(tstart, tstop, event) ~ . - id,
  data = sims$res,
  model = "logit",
  id = sims$res$id,
  by = 1L,
  max_T = 30L,
  Q_0 = diag(1, p + 1L),
  Q = diag(1e-1, p + 1L),
  control = list(method = "GMA"))),

UKF = suppressMessages(ddhazard(
  formula = Surv(tstart, tstop, event) ~ . - id,
  data = sims$res,
  model = "logit",
  id = sims$res$id,
  by = 1L,
  max_T = 30L,
  Q_0 = diag(1, p + 1L),
  Q = diag(1e-1, p + 1L),
  control = list(method = "UKF"))),

  times = 5
))

cat("(n, p) = (", n, ", ", p, ")",
    ". Units is ", sQuote(attr(out, "unit")), "\n", sep = "")

print(out[, c("expr", "lq", "median", "uq")], row.names = FALSE)

cat("\n\n")

invisible()
}

```

Test

```

grid_vals <- expand.grid(
  n = c(250, 1000, 10000),
  p = c(5, 10, 15))

```

```
invisible(
  mapply(get_rune_time_summary, n = grid_vals$n, p = grid_vals$p))
```

```
## (n, p) = (250, 5). Units is 'milliseconds'
##           expr    lq median    uq
## EKF_one_correction_step  57.4   60.2  68.1
## EKF_more_correction_step 115.2  116.1 116.5
##           SMA 192.2  192.2 192.5
##           GMA 174.4  176.5 182.5
##           UKF 275.9  278.6 281.3
##
##
## (n, p) = (1000, 5). Units is 'milliseconds'
##           expr    lq median    uq
## EKF_one_correction_step  189    190 195
## EKF_more_correction_step 531    544 582
##           SMA 566    583 586
##           GMA 259    263 274
##           UKF 468    472 487
##
##
## (n, p) = (10000, 5). Units is 'milliseconds'
##           expr    lq median    uq
## EKF_one_correction_step  393    395 437
## EKF_more_correction_step 913    953 961
##           SMA 2476   2488 2539
##           GMA 767    803 872
##           UKF 2148   2155 2265
##
##
## (n, p) = (250, 10). Units is 'milliseconds'
##           expr    lq median    uq
## EKF_one_correction_step  114    118 128
## EKF_more_correction_step 263    265 268
##           SMA 407    443 453
##           GMA 369    382 394
##           UKF 848    869 881
##
##
## (n, p) = (1000, 10). Units is 'milliseconds'
##           expr    lq median    uq
## EKF_one_correction_step  139    144 145
## EKF_more_correction_step 286    299 302
##           SMA 527    568 587
##           GMA 225    229 245
##           UKF 745    757 761
##
##
## (n, p) = (10000, 10). Units is 'milliseconds'
##           expr    lq median    uq
## EKF_one_correction_step  516    518 554
## EKF_more_correction_step 1039   1113 1313
##           SMA 3385   3427 3612
##           GMA 1236   1249 1311
```

```

##          UKF 3893   3947 3987
##
##
## (n, p) = (250, 15). Units is 'milliseconds'
##      expr   lq median   uq
## EKF_one_correction_step 265   266 279
## EKF_more_correction_step 314   322 329
##          SMA 899   928 956
##          GMA 398   406 407
##          UKF 1982  2080 2154
##
##
## (n, p) = (1000, 15). Units is 'milliseconds'
##      expr   lq median   uq
## EKF_one_correction_step 210   218 224
## EKF_more_correction_step 449   450 458
##          SMA 1014  1020 1220
##          GMA 322   323 329
##          UKF 1458  1506 1549
##
##
## (n, p) = (10000, 15). Units is 'milliseconds'
##      expr   lq median   uq
## EKF_one_correction_step 437   442 536
## EKF_more_correction_step 1013  1072 1078
##          SMA 4759  5813 6034
##          GMA 1158  1212 1354
##          UKF 5366  5462 5529

```

Session info

- R version 3.5.0 (2018-04-23), x86_64-w64-mingw32
- Locale: LC_COLLATE=English_United States.1252, LC_CTYPE=C, LC_MONETARY=English_United States.1252, LC_NUMERIC=C, LC_TIME=English_United States.1252
- Running under: Windows 10 x64 (build 17134)
- Matrix products: default
- Base packages: base, datasets, graphics, grDevices, methods, stats, utils
- Other packages: dynamichazard 0.5.2, microbenchmark 1.4-4, survival 2.41-3
- Loaded via a namespace (and not attached): backports 1.1.2, boot 1.3-20, compiler 3.5.0, digest 0.6.15, evaluate 0.10.1, grid 3.5.0, htmltools 0.3.6, knitr 1.20, lattice 0.20-35, magrittr 1.5, Matrix 1.2-14, parallel 3.5.0, Rcpp 0.12.16, rmarkdown 1.9, rprojroot 1.3-2, splines 3.5.0, stringi 1.1.7, stringr 1.3.0, tools 3.5.0, yaml 2.1.18