

speedtest

Benjamin Christoffersen

2018-05-07

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  59.2   60.2  75
## EKF_more_correction_step 118.6  122.1 128
##           SMA 183.9  190.3 192
##           GMA 170.8  175.7 182
##           UKF 286.5  288.0 293
##
##
## (n, p) = (1000, 5). Units is 'milliseconds'
##           expr    lq median  uq
## EKF_one_correction_step  190    193 201
## EKF_more_correction_step 540    545 553
##           SMA 572    574 580
##           GMA 255    255 260
##           UKF 469    475 495
##
##
## (n, p) = (10000, 5). Units is 'milliseconds'
##           expr    lq median  uq
## EKF_one_correction_step  352    370 375
## EKF_more_correction_step 854    862 863
##           SMA 2348   4216 4224
##           GMA 790    792 796
##           UKF 2184   2192 2202
##
##
## (n, p) = (250, 10). Units is 'milliseconds'
##           expr    lq median  uq
## EKF_one_correction_step  95.5    105 110
## EKF_more_correction_step 223.0    237 241
##           SMA 426.0    427 428
##           GMA 337.3    348 352
##           UKF 813.3    820 824
##
##
## (n, p) = (1000, 10). Units is 'milliseconds'
##           expr    lq median  uq
## EKF_one_correction_step  136    140 142
## EKF_more_correction_step 310    312 315
##           SMA 527    534 770
##           GMA 207    208 212
##           UKF 734    737 749
##
##
## (n, p) = (10000, 10). Units is 'milliseconds'
##           expr    lq median  uq
## EKF_one_correction_step  429    495 507
## EKF_more_correction_step 904    908 926
##           SMA 3085   3252 3334
##           GMA 1023   1038 1062
```

```

##           UKF 3534   3603 3675
##
##
## (n, p) = (250, 15). Units is 'milliseconds'
##           expr   lq median   uq
##   EKF_one_correction_step 305   306 311
##   EKF_more_correction_step 321   339 342
##           SMA 886   1034 1048
##           GMA 416   424 434
##           UKF 2148   2154 2181
##
##
## (n, p) = (1000, 15). Units is 'milliseconds'
##           expr   lq median   uq
##   EKF_one_correction_step 222   222 226
##   EKF_more_correction_step 469   471 475
##           SMA 1065   1105 1216
##           GMA 326   328 335
##           UKF 1485   1497 1502
##
##
## (n, p) = (10000, 15). Units is 'milliseconds'
##           expr   lq median   uq
##   EKF_one_correction_step 399   410 504
##   EKF_more_correction_step 877   937 983
##           SMA 3958   4005 4187
##           GMA 1076   1079 1129
##           UKF 4910   5029 5077

```

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 16299)
- 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