

speedtest

Benjamin Christoffersen

2017-11-18

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  57.8   61.5  61.8
## EKF_more_correction_step 115.3  118.2 126.6
##           SMA 177.2   178.7 179.6
##           GMA 139.9   155.9 165.0
##           UKF 270.3   273.2 286.0
##
##
## (n, p) = (1000, 5). Units is 'milliseconds'
##           expr    lq median    uq
## EKF_one_correction_step  188    197 198
## EKF_more_correction_step  530    538 541
##           SMA  510    511 514
##           GMA  259    267 267
##           UKF  500    502 502
##
##
## (n, p) = (10000, 5). Units is 'milliseconds'
##           expr    lq median    uq
## EKF_one_correction_step  419    501 504
## EKF_more_correction_step 1029   1030 1053
##           SMA 2074   2107 2149
##           GMA  879    968 1007
##           UKF 2336   2344 2371
##
##
## (n, p) = (250, 10). Units is 'milliseconds'
##           expr    lq median    uq
## EKF_one_correction_step  119    122 124
## EKF_more_correction_step  256    262 264
##           SMA  425    426 427
##           GMA  342    345 372
##           UKF  862    906 940
##
##
## (n, p) = (1000, 10). Units is 'milliseconds'
##           expr    lq median    uq
## EKF_one_correction_step  149    152 156
## EKF_more_correction_step  314    318 322
##           SMA  489    492 511
##           GMA  224    230 239
##           UKF  767    774 775
##
##
## (n, p) = (10000, 10). Units is 'milliseconds'
##           expr    lq median    uq
## EKF_one_correction_step  592    598 602
## EKF_more_correction_step  983   1091 1095
##           SMA 2836   2894 2899
##           GMA 1138   1155 1172
```

```

##          UKF 3752   3791 3793
##
##
## (n, p) = (250, 15). Units is 'milliseconds'
##      expr   lq median   uq
##  EKF_one_correction_step 300   300 304
##  EKF_more_correction_step 310   316 318
##      SMA  915   927 938
##      GMA  390   392 401
##      UKF 2071  2088 2117
##
##
## (n, p) = (1000, 15). Units is 'milliseconds'
##      expr   lq median   uq
##  EKF_one_correction_step 236   237 237
##  EKF_more_correction_step 468   469 470
##      SMA  937  1023 1052
##      GMA  348   348 349
##      UKF 1478  1513 1545
##
##
## (n, p) = (10000, 15). Units is 'milliseconds'
##      expr   lq median   uq
##  EKF_one_correction_step 458   536 544
##  EKF_more_correction_step 1061  1080 1091
##      SMA 3852  3874 3875
##      GMA 1180  1266 1290
##      UKF 4838  4861 5084

```

Session info

- R version 3.4.2 (2017-09-28), 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 15063)
- Matrix products: default
- Base packages: base, datasets, graphics, grDevices, methods, stats, utils
- Other packages: dynamichazard 0.5.0, microbenchmark 1.4-2.1, survival 2.41-3
- Loaded via a namespace (and not attached): backports 1.1.0, boot 1.3-20, codetools 0.2-15, colorspace 1.3-2, compiler 3.4.2, data.table 1.10.4-2, digest 0.6.12, evaluate 0.10.1, ggplot2 2.2.1, grid 3.4.2, gtable 0.2.0, htmltools 0.3.6, knitr 1.17, lattice 0.20-35, lazyeval 0.2.0, magrittr 1.5, MASS 7.3-47, Matrix 1.2-11, multcomp 1.4-7, munsell 0.4.3, mvtnorm 1.0-6, parallel 3.4.2, plyr 1.8.4, Rcpp 0.12.13, rlang 0.1.4, rmarkdown 1.6, rprojroot 1.2, sandwich 2.4-0, scales 0.5.0, splines 3.4.2, stringi 1.1.5, stringr 1.2.0, TH.data 1.0-8, tibble 1.3.4, tools 3.4.2, yaml 2.1.14, zoo 1.8-0