speedtest

Benjamin Christoffersen

2017-07-31

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", "cld")], 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 cld
##   EKF_one_correction_step 17.6   17.6 21.3   a
##  EKF_more_correction_step 34.2   34.8 39.2   a
##                       SMA 77.7   82.9 98.4   a
##                       GMA 33.3   34.2 36.4   a
##                       UKF 71.8   73.4 77.8   a
##
##
## (n, p) = (1000, 5). Units is 'milliseconds'
##                       expr    lq median    uq  cld
##   EKF_one_correction_step  29.9   30.5  30.5 a
##  EKF_more_correction_step  48.0   50.1  53.2  b
##                       SMA  57.3   62.3  66.0   c
##                       GMA  37.1   41.0  46.7 ab
##                       UKF 101.0  102.8 111.8    d
##
##
## (n, p) = (10000, 5). Units is 'milliseconds'
##                       expr  lq median  uq  cld
##   EKF_one_correction_step 236    247 251 a
##  EKF_more_correction_step 315    319 416  b
##                       SMA 552    556 573   c
##                       GMA 269    279 291 a
##                       UKF 680    688 707    d
##
##
## (n, p) = (250, 10). Units is 'milliseconds'
##                       expr    lq median    uq  cld
##   EKF_one_correction_step  44.5   48.3  51.0 a
##  EKF_more_correction_step  78.1   78.1  87.3  b
##                       SMA 135.0  147.7 151.1   c
##                       GMA  69.5   71.2  72.2  b
##                       UKF 338.7  350.3 356.4    d
##
##
## (n, p) = (1000, 10). Units is 'milliseconds'
##                       expr    lq median    uq  cld
##   EKF_one_correction_step  37.3   37.7  45.1 a
##  EKF_more_correction_step  59.9   61.9  65.9  b
##                       SMA 108.3  114.9 123.0   c
##                       GMA  46.4   50.6  55.0 ab
##                       UKF 214.7  215.2 224.2    d
##
##
## (n, p) = (10000, 10). Units is 'milliseconds'
##                       expr   lq median   uq  cld
##   EKF_one_correction_step  303    304  309 a
##  EKF_more_correction_step  421    496  504  b
##                       SMA  776    783  784   c
##                       GMA  351    372  373 ab
```

```
##                       UKF 1077   1080 1085    d
##
##
## (n, p) = (250, 15). Units is 'milliseconds'
##                      expr    lq median    uq  cld
##   EKF_one_correction_step 107.9  112.7 113.4  b
##  EKF_more_correction_step  90.3   91.5  92.3 a
##                       SMA 193.7  201.0 201.7   c
##                       GMA  84.1   88.9  91.5 a
##                       UKF 474.7  475.3 480.1    d
##
##
## (n, p) = (1000, 15). Units is 'milliseconds'
##                      expr    lq median    uq  cld
##   EKF_one_correction_step  84.6   90.7  95.4 a
##  EKF_more_correction_step 159.7  166.2 173.9  b
##                       SMA 342.4  353.3 385.6   c
##                       GMA 116.5  119.5 123.6 ab
##                       UKF 543.7  559.7 562.7    d
##
##
## (n, p) = (10000, 15). Units is 'milliseconds'
##                      expr   lq median   uq   cld
##   EKF_one_correction_step  244    244  248 a
##  EKF_more_correction_step  569    574  577   c
##                       SMA  939    943  945    d
##                       GMA  427    442  455  b
##                       UKF 1927   1950 2002     e
```

Session info

- R version 3.4.1 (2017-06-30), `x86_64-w64-mingw32`
- Locale: `LC_COLLATE=English_United Kingdom.1252`, `LC_CTYPE=English_United Kingdom.1252`, `LC_MONETARY=English_United Kingdom.1252`, `LC_NUMERIC=C`, `LC_TIME=English_United Kingdom.1252`
- Running under: `Windows 10 x64 (build 15063)`
- Matrix products: default
- Base packages: base, datasets, graphics, grDevices, methods, stats, utils
- Other packages: dynamichazard 0.3.5, microbenchmark 1.4-2.1, survival 2.41-3
- Loaded via a namespace (and not attached): backports 1.1.0, boot 1.3-19, codetools 0.2-15, colorspace 1.3-2, compiler 3.4.1, data.table 1.10.4, digest 0.6.12, evaluate 0.10, ggplot2 2.2.1, grid 3.4.1, gtable 0.2.0, htmltools 0.3.6, knitr 1.16, lattice 0.20-35, lazyeval 0.2.0, magrittr 1.5, MASS 7.3-47, Matrix 1.2-10, multcomp 1.4-6, munsell 0.4.3, mvtnorm 1.0-6, plyr 1.8.4, Rcpp 0.12.12, rlang 0.1.1, rmarkdown 1.5, rprojroot 1.2, sandwich 2.3-4, scales 0.4.1, splines 3.4.1, stringi 1.1.5, stringr 1.2.0, TH.data 1.0-8, tibble 1.3.3, tools 3.4.1, yaml 2.1.14, zoo 1.8-0