speedtest

Benjamin Christoffersen

2017-06-19

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 24.3   33.0 40.6   a
##  EKF_more_correction_step 41.5   41.8 43.4   a
##                       SMA 85.4   94.7 98.8   a
##                       GMA 36.3   38.0 40.1   a
##                       UKF 84.6   90.4 95.0   a
##
##
## (n, p) = (1000, 5). Units is 'milliseconds'
##                       expr    lq median    uq  cld
##   EKF_one_correction_step  30.4   30.4  35.0 a
##  EKF_more_correction_step  51.1   51.9  56.7  bc
##                       SMA  58.3   62.9  65.6   c
##                       GMA  42.5   49.1  50.5  b
##                       UKF 106.4  110.9 119.1    d
##
##
## (n, p) = (10000, 5). Units is 'milliseconds'
##                       expr  lq median  uq cld
##   EKF_one_correction_step 156    246 252 a
##  EKF_more_correction_step 309    395 396  b
##                       SMA 580    584 596   c
##                       GMA 229    301 312 ab
##                       UKF 650    683 728   c
##
##
## (n, p) = (250, 10). Units is 'milliseconds'
##                       expr    lq median    uq  cld
##   EKF_one_correction_step  40.9   44.2  45.0 a
##  EKF_more_correction_step  72.3   77.6  83.3  b
##                       SMA 131.8  142.3 149.7   c
##                       GMA  68.0   68.0  76.5  b
##                       UKF 336.6  340.3 341.0    d
##
##
## (n, p) = (1000, 10). Units is 'milliseconds'
##                       expr    lq median    uq cld
##   EKF_one_correction_step  41.4   41.7  65.3 a
##  EKF_more_correction_step  54.0   56.0  65.1 a
##                       SMA 128.9  136.7 149.9  b
##                       GMA  48.4   54.5  58.1 a
##                       UKF 229.6  242.5 246.7   c
##
##
## (n, p) = (10000, 10). Units is 'milliseconds'
##                       expr   lq median   uq  cld
##   EKF_one_correction_step  229    294  316 a
##  EKF_more_correction_step  453    484  505  b
##                       SMA  796    798  801   c
##                       GMA  377    391  403 ab
```

```
##                       UKF 1032   1044 1069    d
##
##
## (n, p) = (250, 15). Units is 'milliseconds'
##                      expr    lq median    uq cld
##   EKF_one_correction_step 107.9  119.2 121.6 a
##  EKF_more_correction_step  97.7  127.8 127.9 a
##                       SMA 196.0  215.2 216.2  b
##                       GMA  80.1   94.7  98.5 a
##                       UKF 451.5  456.7 476.2   c
##
##
## (n, p) = (1000, 15). Units is 'milliseconds'
##                      expr    lq median    uq  cld
##   EKF_one_correction_step  89.8   92.9  96.5 a
##  EKF_more_correction_step 146.4  158.3 160.8  b
##                       SMA 342.7  386.1 423.4   c
##                       GMA 116.7  139.8 140.6 ab
##                       UKF 557.0  580.2 581.6    d
##
##
## (n, p) = (10000, 15). Units is 'milliseconds'
##                      expr   lq median   uq  cld
##   EKF_one_correction_step  247    332  351 a
##  EKF_more_correction_step  498    542  550  b
##                       SMA  957    970  972   c
##                       GMA  389    462  478  b
##                       UKF 2050   2063 2123    d
```

Session info

- R version 3.4.0 (2017-04-21), `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 0.3.4, 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 3.4.0, data.table 1.10.4, digest 0.6.12, evaluate 0.10, ggplot2 2.2.1, grid 3.4.0, 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 1.2-9, multcomp 1.4-6, munsell 0.4.3, mvtnorm 1.0-6, plyr 1.8.4, Rcpp 0.12.11, rlang 0.1.1, rmarkdown 1.5, rprojroot 1.2, sandwich 2.3-4, scales 0.4.1, speedglm 0.3-2, splines 3.4.0, stringi 1.1.5, stringr 1.2.0, TH.data 1.0-8, tibble 1.3.3, tools 3.4.0, yaml 2.1.14, zoo 1.8-0