speedtest

Benjamin Christoffersen

2017-06-18

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", "cld")], row.names = FALSE)

  cat("\n\n")

  invisible()
}
```

Test

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

```r
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 18.6   20.1 22.1   a
##  EKF_more_correction_step 31.9   33.3 34.3   a
##                       SMA 76.1   81.0 95.8   a
##                       GMA 33.9   35.3 38.8   a
##                       UKF 74.7   77.2 79.3   a
##
##
## (n, p) = (1000, 5). Units is 'milliseconds'
##                       expr   lq median    uq  cld
##   EKF_one_correction_step 30.4   34.0  34.3 a
##  EKF_more_correction_step 46.8   48.6  50.7  bc
##                       SMA 58.0   59.5  62.7   c
##                       GMA 38.9   40.1  40.5 ab
##                       UKF 98.8  100.6 103.6    d
##
##
## (n, p) = (10000, 5). Units is 'milliseconds'
##                       expr  lq median  uq cld
##   EKF_one_correction_step 148    156 242 a
##  EKF_more_correction_step 395    404 407  b
##                       SMA 481    510 584   c
##                       GMA 196    202 278 a
##                       UKF 566    574 669   c
##
##
## (n, p) = (250, 10). Units is 'milliseconds'
##                       expr    lq median    uq  cld
##   EKF_one_correction_step  41.3   47.2  50.4 a
##  EKF_more_correction_step  73.2   75.4  79.6  b
##                       SMA 131.5  139.3 140.3   c
##                       GMA  67.4   67.9  68.0  b
##                       UKF 334.9  335.3 341.0    d
##
##
## (n, p) = (1000, 10). Units is 'milliseconds'
##                       expr    lq median    uq  cld
##   EKF_one_correction_step  36.8   37.2  40.3 a
##  EKF_more_correction_step  56.5   57.3  61.2  b
##                       SMA 112.3  113.3 119.2   c
##                       GMA  47.7   50.9  56.1  b
##                       UKF 212.9  217.9 218.7    d
##
##
## (n, p) = (10000, 10). Units is 'milliseconds'
##                       expr   lq median   uq  cld
##   EKF_one_correction_step  231    234  234 a
##  EKF_more_correction_step  359    361  474  b
##                       SMA  794    796  798   c
##                       GMA  330    383  383  b
```

```
##                       UKF 1061   1076 1109    d
##
##
## (n, p) = (250, 15). Units is 'milliseconds'
##                      expr    lq median    uq cld
##   EKF_one_correction_step 104.7  117.1 118.3 a
##  EKF_more_correction_step  88.0   88.9  93.9 a
##                       SMA 181.8  184.4 196.0  b
##                       GMA  95.8   96.4 100.8 a
##                       UKF 482.0  483.8 528.4   c
##
##
## (n, p) = (1000, 15). Units is 'milliseconds'
##                      expr    lq median    uq  cld
##   EKF_one_correction_step  78.2   79.9  85.2 a
##  EKF_more_correction_step 141.2  148.2 156.2  b
##                       SMA 332.4  418.5 435.2   c
##                       GMA 116.1  123.4 124.2 ab
##                       UKF 525.5  526.8 530.4    d
##
##
## (n, p) = (10000, 15). Units is 'milliseconds'
##                      expr   lq median   uq   cld
##   EKF_one_correction_step  310    329  336 a
##  EKF_more_correction_step  537    569  571   c
##                       SMA  970   1039 1043    d
##                       GMA  447    466  469  b
##                       UKF 1914   1978 1980     e
```

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 0.3.3, 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