speedtest

Benjamin Christoffersen

2017-07-19

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))
```

```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 19.8   20.9 22.1   a
##  EKF_more_correction_step 35.2   40.4 41.5   a
##                       SMA 87.8   90.2 99.7   a
##                       GMA 35.6   35.9 36.0   a
##                       UKF 80.9   86.9 88.6   a
##
##
## (n, p) = (1000, 5). Units is 'milliseconds'
##                       expr   lq median    uq  cld
##   EKF_one_correction_step 34.9   35.0  37.1 a
##  EKF_more_correction_step 46.9   51.7  53.4  bc
##                       SMA 58.1   58.8  59.6   c
##                       GMA 39.3   39.5  43.3 ab
##                       UKF 96.4  100.2 109.0    d
##
##
## (n, p) = (10000, 5). Units is 'milliseconds'
##                       expr  lq median  uq cld
##   EKF_one_correction_step 153    252 255 a
##  EKF_more_correction_step 298    310 405  b
##                       SMA 478    576 576   c
##                       GMA 207    207 304 ab
##                       UKF 603    660 677   c
##
##
## (n, p) = (250, 10). Units is 'milliseconds'
##                       expr    lq median    uq  cld
##   EKF_one_correction_step  40.4   41.2  44.8 a
##  EKF_more_correction_step  77.6   78.6  83.1  b
##                       SMA 131.6  135.9 141.2   c
##                       GMA  72.4   79.3  83.4  b
##                       UKF 341.0  341.8 344.7    d
##
##
## (n, p) = (1000, 10). Units is 'milliseconds'
##                       expr    lq median    uq  cld
##   EKF_one_correction_step  36.4   37.8  38.3 a
##  EKF_more_correction_step  53.3   53.6  55.8  b
##                       SMA 119.0  123.3 128.5   c
##                       GMA  48.2   48.4  50.0  b
##                       UKF 210.5  211.1 212.9    d
##
##
## (n, p) = (10000, 10). Units is 'milliseconds'
##                       expr   lq median   uq  cld
##   EKF_one_correction_step  308    309  316 a
##  EKF_more_correction_step  364    370  476  b
##                       SMA  762    763  768   c
##                       GMA  279    366  374 ab
```

```
##                       UKF 1068   1068 1078    d
##
##
## (n, p) = (250, 15). Units is 'milliseconds'
##                      expr    lq median  uq  cld
##   EKF_one_correction_step  96.9   97.0 106  b
##  EKF_more_correction_step  93.5   97.1  98  b
##                       SMA 182.3  183.7 187   c
##                       GMA  75.2   78.8  87 a
##                       UKF 451.0  455.6 457    d
##
##
## (n, p) = (1000, 15). Units is 'milliseconds'
##                      expr    lq median    uq  cld
##   EKF_one_correction_step  81.9   83.9  87.4 a
##  EKF_more_correction_step 151.3  154.5 158.9  b
##                       SMA 327.3  371.2 407.2   c
##                       GMA 118.0  118.6 124.0 ab
##                       UKF 532.0  533.6 537.5    d
##
##
## (n, p) = (10000, 15). Units is 'milliseconds'
##                      expr   lq median   uq  cld
##   EKF_one_correction_step  239    331  347 a
##  EKF_more_correction_step  537    550  562  b
##                       SMA  940    961 1073   c
##                       GMA  439    445  460  b
##                       UKF 1909   1978 2061    d
```

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 0.12.11, rlang 0.1.1, rmarkdown 1.5, rprojroot 1.2, sandwich 2.3-4, scales 0.4.1, speedglm 0.3-2, splines 3.4.1, 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