speedtest

Benjamin Christoffersen

2017-08-05

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  72.2   78.3  79  a
##  EKF_more_correction_step 130.4  130.8 132  ab
##                       SMA 166.4  175.7 176   b
##                       GMA 184.3  184.9 190  ab
##                       UKF 204.6  206.4 209  ab
##
##
## (n, p) = (1000, 5). Units is 'milliseconds'
##                       expr  lq median  uq  cld
##   EKF_one_correction_step 122    125 129 a
##  EKF_more_correction_step 305    306 312  b
##                       SMA 502    505 523    d
##                       GMA 321    324 331  b
##                       UKF 418    421 423   c
##
##
## (n, p) = (10000, 5). Units is 'milliseconds'
##                       expr   lq median   uq   cld
##   EKF_one_correction_step  510    512  517 a
##  EKF_more_correction_step 1045   1060 1095   c
##                       SMA 2012   2100 2132    d
##                       GMA  756    855  860  b
##                       UKF 2171   2269 2309     e
##
##
## (n, p) = (250, 10). Units is 'milliseconds'
##                       expr  lq median  uq   cld
##   EKF_one_correction_step 121    126 141 a
##  EKF_more_correction_step 284    291 293  b
##                       SMA 417    421 424    d
##                       GMA 370    374 381   c
##                       UKF 556    570 582     e
##
##
## (n, p) = (1000, 10). Units is 'milliseconds'
##                       expr  lq median  uq  cld
##   EKF_one_correction_step 115    115 115 a
##  EKF_more_correction_step 231    239 239  b
##                       SMA 486    507 516   c
##                       GMA 246    250 256  b
##                       UKF 601    604 606    d
##
##
## (n, p) = (10000, 10). Units is 'milliseconds'
##                       expr   lq median   uq  cld
##   EKF_one_correction_step  616    632  645 a
##  EKF_more_correction_step 1197   1213 1264  b
##                       SMA 3048   3147 3210   c
##                       GMA 1188   1220 1230  b
```

```
##                       UKF 3404   3412 3478    d
##
##
## (n, p) = (250, 15). Units is 'milliseconds'
##                      expr   lq median   uq  cld
##   EKF_one_correction_step  310    314  321 a
##  EKF_more_correction_step  356    356  358 a
##                       SMA  871    939  973   c
##                       GMA  422    437  444  b
##                       UKF 1145   1148 1152    d
##
##
## (n, p) = (1000, 15). Units is 'milliseconds'
##                      expr   lq median   uq cld
##   EKF_one_correction_step  185    193  194 a
##  EKF_more_correction_step  358    358  375  b
##                       SMA  994   1081 1101   c
##                       GMA  409    409  411  b
##                       UKF 1075   1078 1085   c
##
##
## (n, p) = (10000, 15). Units is 'milliseconds'
##                      expr   lq median   uq   cld
##   EKF_one_correction_step  501    507  590 a
##  EKF_more_correction_step 1250   1261 1272  b
##                       SMA 3990   4037 4092    d
##                       GMA 1356   1371 1448   c
##                       UKF 4271   4295 4425     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