

speedtest

Benjamin Christoffersen

2018-05-23

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")], 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
## EKF_one_correction_step  60   60.9  69.5
## EKF_more_correction_step 111  113.2 115.7
##           SMA 188  188.2 204.3
##           GMA 178  180.7 181.8
##           UKF 268  273.8 274.6
##
##
## (n, p) = (1000, 5). Units is 'milliseconds'
##           expr  lq median  uq
## EKF_one_correction_step 187   189 193
## EKF_more_correction_step 592   597 647
##           SMA 617   628 658
##           GMA 258   269 289
##           UKF 485   487 505
##
##
## (n, p) = (10000, 5). Units is 'milliseconds'
##           expr  lq median  uq
## EKF_one_correction_step 365   374 403
## EKF_more_correction_step 927   951 966
##           SMA 2386  2417 2513
##           GMA 794   811 924
##           UKF 2382  2445 2484
##
##
## (n, p) = (250, 10). Units is 'milliseconds'
##           expr  lq median  uq
## EKF_one_correction_step 102   103 107
## EKF_more_correction_step 224   228 256
##           SMA 417   417 472
##           GMA 327   336 344
##           UKF 790   794 798
##
##
## (n, p) = (1000, 10). Units is 'milliseconds'
##           expr  lq median  uq
## EKF_one_correction_step 132   136 141
## EKF_more_correction_step 294   298 301
##           SMA 505   540 565
##           GMA 207   210 210
##           UKF 762   781 783
##
##
## (n, p) = (10000, 10). Units is 'milliseconds'
##           expr  lq median  uq
## EKF_one_correction_step 455   533 546
## EKF_more_correction_step 953   984 996
##           SMA 3136  3289 4840
##           GMA 1060  1068 1127
```

```

##          UKF 3564   3709 3808
##
##
## (n, p) = (250, 15). Units is 'milliseconds'
##      expr   lq median   uq
## EKF_one_correction_step 298   327 337
## EKF_more_correction_step 323   334 349
##          SMA  904   1026 1068
##          GMA  426   455 475
##          UKF 2182   2297 2441
##
##
## (n, p) = (1000, 15). Units is 'milliseconds'
##      expr   lq median   uq
## EKF_one_correction_step 223   230 231
## EKF_more_correction_step 454   454 492
##          SMA 1128   1335 1374
##          GMA  336   346 366
##          UKF 1576   1585 1673
##
##
## (n, p) = (10000, 15). Units is 'milliseconds'
##      expr   lq median   uq
## EKF_one_correction_step 406   422 527
## EKF_more_correction_step 947   986 1002
##          SMA 4089   4117 5636
##          GMA 1085   1124 1175
##          UKF 4793   4932 4955

```

Session info

- R version 3.5.0 (2018-04-23), x86_64-w64-mingw32
- Locale: LC_COLLATE=English_United States.1252, LC_CTYPE=C, LC_MONETARY=English_United States.1252, LC_NUMERIC=C, LC_TIME=English_United States.1252
- Running under: Windows 10 x64 (build 17134)
- Matrix products: default
- Base packages: base, datasets, graphics, grDevices, methods, stats, utils
- Other packages: dynamichazard 0.5.2, microbenchmark 1.4-4, survival 2.41-3
- Loaded via a namespace (and not attached): backports 1.1.2, boot 1.3-20, compiler 3.5.0, digest 0.6.15, evaluate 0.10.1, grid 3.5.0, htmltools 0.3.6, knitr 1.20, lattice 0.20-35, magrittr 1.5, Matrix 1.2-14, parallel 3.5.0, Rcpp 0.12.16, rmarkdown 1.9, rprojroot 1.3-2, splines 3.5.0, stringi 1.1.7, stringr 1.3.0, tools 3.5.0, yaml 2.1.18