VARS in R

Arnald Puy

```r
# PRELIMINARY FUNCTIONS -------------------------------------------------------

# Function to read in all required packages in one go:
loadPackages <- function(x) {
  for(i in x) {
    if(!require(i, character.only = TRUE)) {
      install.packages(i, dependencies = TRUE)
      library(i, character.only = TRUE)
    }
  }
}

# Install development version of sensobol
remotes::install_github("arnaldpuy/sensobol")

# Load the packages
loadPackages(c("tidyverse", "sensobol"))



# Create custom theme
theme_AP <- function() {
  theme_bw() +
    theme(panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          legend.background = element_rect(fill = "transparent",
                                           color = NA),
          legend.key = element_rect(fill = "transparent",
                                    color = NA))
}

# Set checkpoint

dir.create(".checkpoint")
library("checkpoint")

checkpoint("2020-03-09",
           R.version ="3.6.1",
           checkpointLocation = getwd())
```

```r
# FUNCTION TO CREATE STAR-VARS ------------------------------------------------

vars_matrices <- function(N, params, h) {
  out <- center <- sections <- A <- B <- AB <- X <- out <- list()
  mat <- randtoolbox::sobol(n = N, dim = length(params))
  for(i in 1:nrow(mat)) {
    center[[i]] <- mat[i, ]
    sections[[i]] <- sapply(center[[i]], function(x) {
      all <- seq(x %% h, 1, h)
      non.zeros <- all[all!= 0] # Remove zeroes
      })
    B[[i]] <- sapply(1:ncol(mat), function(x)
      sections[[i]][, x][!sections[[i]][, x] %in% center[[i]][x]])
    A[[i]] <- matrix(center[[i]], nrow = nrow(B[[i]]), ncol = length(center[[i]]), byrow = TRU
    X[[i]] <- rbind(A[[i]], B[[i]])
    for(j in 1:ncol(A[[i]])) {
      AB[[i]] <- A[[i]]
      AB[[i]][, j] <- B[[i]][, j]
      X[[i]] <- rbind(X[[i]], AB[[i]])
    }
    AB[[i]] <- X[[i]][(2 * nrow(B[[i]]) + 1):nrow(X[[i]]), ]
    out[[i]] <- rbind(unname(center[[i]]), AB[[i]])
  }
  return(do.call(rbind, out))
}

# Function to cut by size
CutBySize <- function(m, block.size, nb = ceiling(m / block.size)) {
  int <- m / nb
  upper <- round(1:nb * int)
  lower <- c(1, upper[-nb] + 1)
  size <- c(upper[1], diff(upper))
  cbind(lower, upper, size)
}

# Function to compute VARS-TI
vars_ti <- function(Y, N, params, h) {
  n.cross.points <- length(params) * ((1 / h) - 1) + 1
  index.centers <- seq(1, length(Y), n.cross.points)
  mat <- matrix(Y[-index.centers], ncol = N)
  indices <- CutBySize(nrow(mat), nb = length(params))
  out <- list()
  for(i in 1:nrow(indices)) {
    out[[i]] <- mat[indices[i, "lower"]:indices[i, "upper"], ]
  }
  d <- lapply(1:length(params), function(x)
    lapply(1:ncol(out[[x]]), function(j) {
      da <- c(out[[x]][, j][1],
              rep(out[[x]][, j][-c(1, length(out[[x]][, j]))], each = 2),
              out[[x]][, j][length(out[[x]][, j])])
    }))
  out <- lapply(d, function(x) lapply(x, function(y) matrix(y, nrow = length(y) / 2, byrow = T
  variogr <- unlist(lapply(out, function(x)
    lapply(x, function(y) 1 / 2 * mean(y[, 1] - y[, 2]) ^ 2)) %>%
      lapply(., function(x) Rfast::colmeans(do.call(rbind, x))))
  covariogr <- unlist(lapply(out, function(x)
    lapply(x, function(y) cov(y[, 1], y[, 2]))) %>%
      lapply(., function(x) Rfast::colmeans(do.call(rbind, x))))
  VY <- var(Y[index.centers])
  output <- (variogr + covariogr) / VY
  return(output)
}
```

```
# DEFINE THE SETTINGS FOR A STAR-VARS SAMPLE MATRIX ---------------------------

N <- 200 # Star centers
h <- 0.1 # h step
```

```
# VARS-TO FOR THE ISHIGAMI FUNCTION -------------------------------------------

params <- paste("X", 1:3, sep = "")
mat <- vars_matrices(N = N, params = params, h = h)
Y <- sensobol::ishigami_Fun(mat)
output <- vars_ti(Y = Y, N = N, params = params, h = h)
print(round(output, 3))
```

```
## [1] 0.960 0.000 0.215
```

```
# VARS-TO FOR THE SOBOL' G FUNCTION -------------------------------------------

params <- paste("X", 1:8, sep = "")
mat <- vars_matrices(N = N, params = params, h = h)
Y <- sensobol::sobol_Fun(mat)
output <- vars_ti(Y = Y, N = N, params = params, h = h)
print(round(output, 3))
```

```
## [1] 0.565 0.171 0.024 0.007 0.000 0.000 0.000 0.000
```

```
# VARS-TO FOR THE MORRIS FUNCTION -------------------------------------------

params <- paste("X", 1:20, sep = "")
mat <- vars_matrices(N = N, params = params, h = h)
Y <- sensitivity::morris.fun(mat)
```

```
## Registered S3 method overwritten by 'sensitivity':
##   method    from
##   print.src dplyr
```

```r
output <- vars_ti(Y = Y, N = N, params = params, h = h)
print(round(output, 3))
```

```
##  [1] 0.244 0.247 0.082 0.263 0.078 0.088 0.047 0.159 0.158 0.117 0.002 0.006
## [13] 0.003 0.003 0.002 0.002 0.003 0.003 0.001 0.004
```