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", "data.table"))



# 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 ------------------------------------------------

# CROSS-SECTIONS WITHOUT STAR CENTER-----------------------------

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 = TRUE)
    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)
    mean(0.5 * (y[, 1] - y[, 2]) ^ 2))) %>%
      lapply(., function(x) do.call(rbind, x)) %>%
      lapply(., mean))
  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])
  Ti <- (variogr + covariogr) / VY
  output <- data.table::data.table(Ti)
  output[, `:=`(parameters = params)]
  return(output)
}


# CROSS-SECTIONS WITH STAR CENTER ----------------------------------

vars_matrices_NEW <- 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(mat[i, ], function(x) {
      all <- seq(x %% h, 1, h)
      non.zeros <- all[all!= 0] # Remove zeroes
    })
    A[[i]] <- matrix(mat[i, ], nrow = nrow(sections[[i]]),
                     ncol = ncol(mat), byrow = TRUE)
    X[[i]] <- rbind(A[[i]], sections[[i]])
    for(j in 1:ncol(A[[i]])) {
      AB[[i]] <- A[[i]]
      AB[[i]][, j] <- sections[[i]][, j]
      X[[i]] <- rbind(X[[i]], AB[[i]])
    }
    AB[[i]] <- X[[i]][(2 * nrow(sections[[i]]) + 1):nrow(X[[i]]), ]
  }
  star.vars <- do.call(rbind, AB)
  tmp <- lapply(1:N, function(x)
    rowSums(star.vars == mat[x, ][col(star.vars)]) == ncol(star.vars))
  indices.star.output <- unlist(lapply(1:N, function(x) which(tmp[[x]] == TRUE)[1]))
  return(list(star.vars, indices.star.output))
}

pairs_tmp <- function(x, h) {
  da <- list()
  for(i in 1:((1/h) - 1)) {
    da[[i]] <- c(x[i], x[i+1])
  }
  out <- do.call(rbind, da)
  return(out)
}

pairs_fun <- function(x, h) {
  da <- list()
  for(j in 1:ncol(x)) {
    da[[j]] <- pairs_tmp(x[, j], h = h)
  }
  return(da)
}

vars_ti_NEW <- function(Y, N, indices.var, params, h) {
  mat <- matrix(Y, 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(out, function(x) pairs_fun(x, h = h))
  variogr <- unlist(lapply(d, function(x) lapply(x, function(y)
    mean(0.5 * (y[, 1] - y[, 2]) ^ 2))) %>%
      lapply(., function(x) do.call(rbind, x)) %>%
      lapply(., mean))
  covariogr <- unlist(lapply(d, function(x)
    lapply(x, function(y) cov(y[, 1], y[, 2]))) %>%
      lapply(., function(x) Rfast::colmeans(do.call(rbind, x))))
  VY <- var(Y[indices.var])
  Ti <- (variogr + covariogr) / VY
  output <- data.table::data.table(Ti)
  output[, `:=`(parameters = params)]
  return(output)
}
```

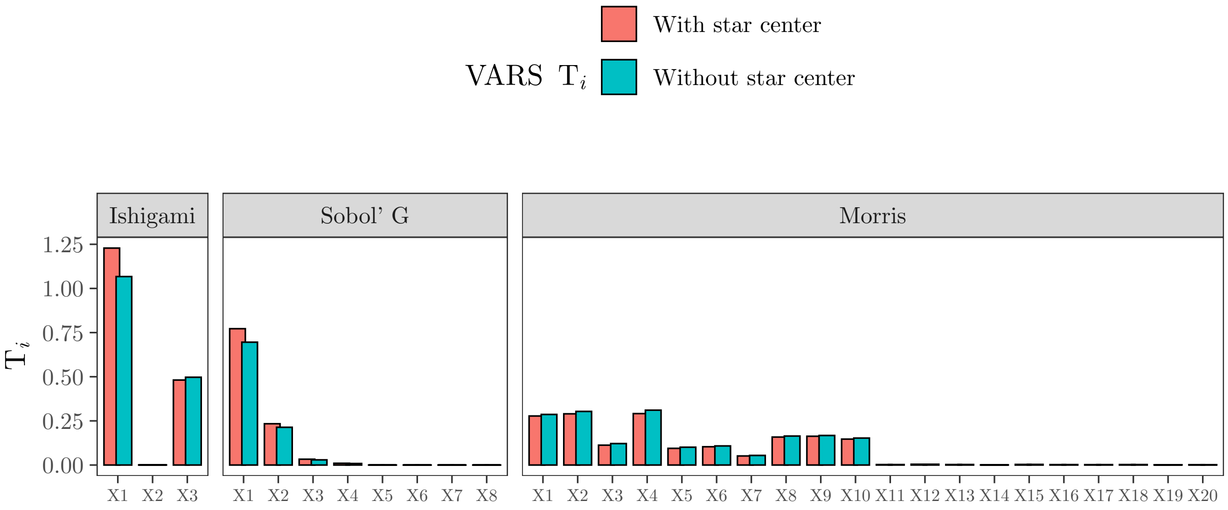
```
# COMPARE RESULTS  ------------------------------------------------------------

vars_type <- c("With", "Without")
test_functions <- c("Ishigami", "Sobol' G", "Morris")
```

```
N <- 200
h <- 0.2

# Run model
mat <- Y <- ind <- list()
for(i in test_functions) {
  for(j in vars_type) {
    if(i == "Ishigami") {
      k <- 3
      modelRun <- sensobol::ishigami_Fun
    } else if(i == "Sobol' G") {
      k <- 8
      modelRun <- sensobol::sobol_Fun
    } else if(i == "Morris") {
      k <- 20
      modelRun <- sensitivity::morris.fun
    }
    if(j == "With") {
      mat[[i]][[j]] <- vars_matrices_NEW(N = N,
                                         params = paste("X", 1:k, sep = ""),
                                         h = h)
      Y[[i]][[j]] <- modelRun(mat[[i]][[j]][[1]])
      ind[[i]][[j]] <- vars_ti_NEW(Y = Y[[i]][[j]],
                                   N = N,
                                   indices.var = mat[[i]][[j]][[2]],
                                   params = paste("X", 1:k, sep = ""),
                                   h = h)
    } else if(j == "Without") {
      mat[[i]][[j]] <- vars_matrices(N = N,
                                     params = paste("X", 1:k, sep = ""),
                                     h = h)
      Y[[i]][[j]] <- modelRun(mat[[i]][[j]])
      ind[[i]][[j]] <- vars_ti(Y = Y[[i]][[j]],
                                   N = N,
                                   params = paste("X", 1:k, sep = ""),
                                   h = h)
    }
  }
}
```

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

```
# PLOT RESULTS ----------------------------------------------------------------

lapply(ind, function(x) rbindlist(x, idcol = "Type")) %>%
  rbindlist(., idcol = "Function") %>%
```

```r
.[, parameters:= factor(parameters, levels = paste("X", 1:20, sep = ""))] %>%
.[, Function:= factor(Function, levels = test_functions)] %>%
ggplot(., aes(parameters, Ti, fill = Type)) +
geom_bar(stat = "identity",
         position = position_dodge(0.7),
         color = "black") +
scale_fill_discrete(name = expression(paste("VARS "~T[italic(i)])),
                    labels = c("With star center", "Without star center")) +
facet_grid(~Function,
           scales = "free_x",
           space = "free_x") +
labs(x = "",
     y = expression(T[italic(i)])) +
theme_AP() +
theme(axis.text.x = element_text(size = 6.5),
      legend.position = "top") +
guides(fill = guide_legend(nrow = 3,
                           byrow = TRUE))
```


VARS T_i
🟥 With star center
🟦 Without star center
Ishigami
Sobol' G
Morris
T_i
X1 X2 X3
X1 X2 X3 X4 X5 X6 X7 X8
X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 X17 X18 X19 X20