The battle of variance-based sensitivity estimators

Arnald Puy, Samuele Lo Piano, Andrea Saltelli

Contents

```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)
    }
  }
}

# Load the packages
loadPackages(c("Rcpp", "tidyverse", "parallel", "foreach", "doParallel",
               "Rfast", "data.table", "scales"))
# 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("2020-01-23",
           R.version ="3.6.1",
           checkpointLocation = getwd())
```

0.1 The metafunction

We use Becker (2019)'s metafunction approach, and define a metafunction including ten different functions. The code and the functions are shown below:

```r
# CREATE METAFUNCTION -----------------------------------------------------

function_list <- list(
  Linear = function(x) x,
  Quadratic = function(x) x ^ 2,
  Cubic = function(x) x ^ 3,
  Exponential = function(x) exp(1) ^ x / (exp(1) - 1),
  Periodic = function(x) sin(2 * pi * x) / 2,
```

```
  Discontinuous = function(x) ifelse(x > 0.5, 1, 0),
  Non.monotonic = function(x) 4 * (x - 0.5) ^ 2,
  Inverse = function(x) (10 - 1 / 1.1) ^ -1 * (x + 0.1) ^ - 1,
  No.effect = function(x) x * 0,
  Trigonometric = function(x) cos(x)
)
```

```
# PLOT METAFUNCTION ------------------------------------------------------------

ggplot(data.frame(x = runif(100)), aes(x)) +
  map(1:length(function_list), function(nn) {
    stat_function(fun = function_list[[nn]],
                  geom = "line",
                  aes_(color = factor(names(function_list[nn])),
                       linetype = factor(names(function_list[nn]))))
  }) +
  labs(color= "Function", linetype = "Function",
       x = expression(italic(x)),
       y = expression(italic(y))) +
  theme_AP() +
  theme(legend.position = "right")
```

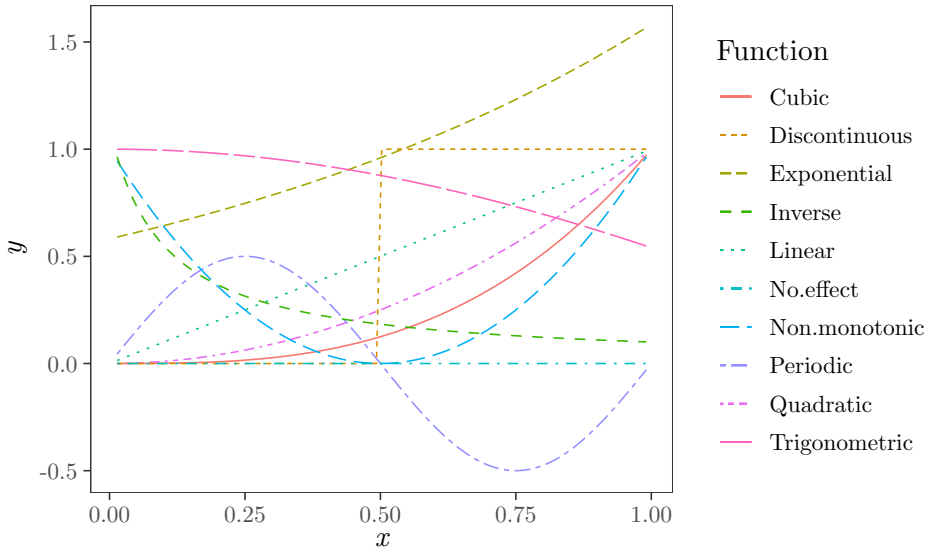

Figure 1: Functions used in the metafunction of Becker (2019).

References

Becker, William. 2019. “Sensitivity analysis on a shoestring : screening model inputs at low sample size.”