Introduction

The package glober provides two tools to estimate the function $f$ in the following nonparametric regression model:

$$Y_i = f(x_i) + \varepsilon_i, \quad 1 \leq i \leq n,$$

where the $\varepsilon_i$ are i.i.d centered random variables of variance $\sigma^2$, the $x_i$ are observation points which belong to a compact set $S \subseteq \mathbb{R}^d$, $d = 1$ or $2$ and $n$ is the total number of observations. This estimation is performed using the GLOBER approach described in [1]. This method consists in estimating $f$ by approximating it with a linear combination of B-splines, where their knots are selected adaptively using the Generalized Lasso proposed by [2], since they can be seen as changes in the derivatives of the function to estimate. We refer the reader to [1] for further details.

Estimation of $f$ in the one-dimensional case ($d = 1$)

In the following, we apply our method to a function of one input variable $f_1$. This function is defined as a linear combination of quadratic B-splines with the set of knots $t = (0.1, 0.27, 0.745)$ and $\sigma = 0.1$ in (1).

Description of the dataset

We load the dataset of observations with $n = 70$ provided within the package $(x_1, \ldots, x_{70})$:

```r
## --- Loading the values of the input variable --- ##
data('x_1D')
```

and $(Y_1, \ldots, Y_{70})$:

```r
## --- Loading the corresponding noisy values of the response variable --- ##
data('y_1D')
```

We load the dataset containing the values of the input variable $(x_1, \ldots, x_N)$ for which an estimation of $f_1$ is sought. They correspond to the observation points as well as additional points where $f_1$ has not been observed. Here, $N = 201$. In order to have a better idea of the underlying function $f_1$, we load the corresponding evaluations of $f_1$ at these input values.

```r
## --- Loading the values of the input variable for which an estimation
## of f_1 is required --- ##
data('xpred_1D')

## --- Loading the corresponding evaluations to plot the function --- ##
data('f_1D')
```

We can visualize it for 201 input values by using the ggplot2 package:

```r
## -- Building dataframes to plot -- ##
data_1D = data.frame(x = xpred_1D, f = f_1D)
obs_1D = data.frame(x = x_1D, y = y_1D)
real.knots = c(0.1, 0.27, 0.745)
```
The vertical dashed lines represent the real knots \( t \) implied in the definition of \( f_1 \), the red curve describes the true underlying function \( f_1 \) to estimate and the blue crosses are the observation points.

**Application of glober.1d to estimate \( f_1 \)**

The `glober.1d` function of the `glober` package is applied by using the following arguments: the input values \((x_i)_{1 \leq i \leq n} \) (\( x \)), the corresponding \((Y_i)_{1 \leq i \leq n} \) (\( y \)), \( N \) input values \( \{x_1, \ldots, x_N\} \) for which \( f_1 \) has to be estimated (\( x_{\text{pred}} \)) and the order of the B-spline basis used to estimate \( f_1 \) (\( \text{ord} \)).

```r
res = glober.1d(x = x_1D, y = y_1D, xpred = xpred_1D, ord = 3, parallel = FALSE)
```

Additional arguments can also be used in this function:

- `parallel`: Logical, if set to TRUE then a parallelized version of the code is used. The default value is FALSE.
- `nb.Cores`: Numerical, it represents the number of cores used for parallelization, if parallel is set to TRUE.

The resulting outputs are the following:

- `festimated`: the estimated values of \( f_1 \).
- `knotSelect`: the selected knots used in the definition of the B-splines of the GLOBER estimator.
- `rss`: Residual sum-of-squares (RSS) of the model defined as: \( \sum_{k=1}^{n}(Y_i - \hat{f}_1(x_i))^2 \), where \( \hat{f}_1 \) is the estimator of \( f_1 \).
- `rsq`: R-squared of the model, calculated as \( 1 - \frac{\text{RSS}}{\text{TSS}} \) where TSS is the total sum-of-squares of the model defined as \( \sum_{k=1}^{n}(Y_i - \bar{Y})^2 \) with \( \bar{Y} = \frac{\sum_{i=1}^{n}Y_i}{n} \).

Thus, we can print the estimated values corresponding to the input values \( \{x_1, \ldots, x_N\} \):

```r
fhat = res$festimated
head(fhat)
```

```r
## [1] -0.02579931 -0.26804301 -0.49284982 -0.70021972 -0.89015272 -1.06264882
```

The value of the Residual Sum-of-square:

```r
res$rss
```

```r
## [1] 40.91661
```

The value of the R-squared:

```r
res$rsq
```

```r
## [1] 0.9970843
```

We can get the set of the estimated knots \( \hat{t} \):
Finally, we can display the estimation of $f_1$ by using the `ggplot2` package:

```r
## Dataframe of selected knots ##
idknots = which(xpred_1D %in% knots.set)
yknots = f_1D[idknots]
data_knots = data.frame(x.knots = knots.set, y.knots = yknots)

## Dataframe of the estimation ##
data_res = data.frame(xpred = xpred_1D, fhat = fhat)

plot_1D = ggplot(data_1D, aes(xpred_1D, f_1D)) +
  geom_line(color = 'red') +
  geom_line(data = data_1D, aes(x = xpred_1D, y = fhat), color = "black") +
  geom_vline(xintercept = real.knots, linetype = 'dashed', color = 'grey27') +
  geom_point(aes(x, y), data = obs_1D, shape = 4, color = "blue", size = 4)+
  geom_point(aes(x = x.knots, y = y.knots), data = data_knots, shape = 19, color = "blue",
             size = 4)+
  xlab('x') +
  ylab('y') +
  theme_bw() +
  theme(axis.title.x = element_text(size = 20), axis.title.y = element_text(size = 20),
        axis.text.x = element_text(size = 19),
        axis.text.y = element_text(size = 19))

plot_1D
```

The vertical dashed lines represent the real knots $t$ implied in the definition of $f_1$, the red curve describes the true underlying function $f_1$ to estimate, the black curve corresponds to the estimation with GLOBER, the blue crosses are the observation points and the blue bullets are the observation points chosen as estimated knots $\hat{t}$.

**Estimation of $f$ in the two-dimensional case ($d = 2$)**

In the following, we apply our method to a function of two input variables $f_2$. This function is defined as a linear combination of tensor products of quadratic univariate B-splines with the sets of knots $t_1 = (0.24, 0.545)$ and $t_2 = (0.395, 0.645)$ and $\sigma = 0.01$ in (1).
Description of the dataset

We load the dataset of observations with \( n = 100 \), provided within the package \((x_1, \ldots, x_{100})\)

```r
## --- Loading the values of the input variables --- ##
data('x_2D')
head(x_2D)
```

```
## Var1 Var2
## [1,] 0.005 0.005
## [2,] 0.005 0.385
## [3,] 0.005 0.390
## [4,] 0.005 0.395
## [5,] 0.005 0.640
## [6,] 0.005 0.645
```

and \((Y_1, \ldots, Y_{100})\):

```r
## --- Loading the corresponding noisy values of the response variable --- ##
data('y_2D')
```

We load the dataset containing the values of the input variables \(\{x_1, \ldots, x_N\}\) for which an estimation of \(f_2\) is sought. They correspond to the observation points as well as additional points where \(f_2\) has not been observed. Here, \(N = 10000\). In order to have a better idea of the underlying function \(f_2\), we load the corresponding evaluations of \(f_2\) at these input values.

```r
## --- Loading the values of the input variables for which an estimation 
## of f_2 is required --- ##
data('xpred_2D')
head(xpred_2D)
```

```
## Var1 Var2
## [1,] 0 0.000
## [2,] 0 0.005
## [3,] 0 0.015
## [4,] 0 0.035
## [5,] 0 0.050
## [6,] 0 0.080
```

```r
## --- Loading the corresponding evaluations to plot the function --- ##
data('f_2D')
```

We can visualize it for 10000 input values by using the `plot3D` package.
Application of glober.2d to estimate $f_2$

The glober.2d function of the glober package is applied by using the following arguments: the input values $(x_i)_{1 \leq i \leq n}$ (x), the corresponding $(Y_i)_{1 \leq i \leq n}$ (y), $N$ input values $\{x_1, \ldots, x_N\}$ for which $f_2$ has to be estimated (xpred) and the order of the B-spline basis used to estimate $f_2$ (ord).

```r
res = glober.2d(x = x_2D, y = y_2D, xpred = xpred_2D, ord = 3, parallel = FALSE)
```

Additional arguments can also be used in this function:
- **parallel**: Logical, if TRUE then a parallelized version of the code is used. Default is FALSE.
- **nb.Cores**: Numerical, it corresponds to the number of cores used for parallelization, if parallel is set to TRUE.

Outputs:
- **festimated**: the estimated values of $f_2$.
- **knotSelec**: the selected knots used in the definition of the B-splines of the GLOBER estimator.
- **rss**: Residual sum-of-squares (RSS) of the model defined as: $\sum_{k=1}^{n}(Y_i - \hat{f}_2(x_i))^2$, where $\hat{f}_2$ is the estimator of $f_2$.
- **rsq**: R-squared of the model, calculated as $1 - \frac{RSS}{TSS}$ where TSS is the total sum-of-squares of the model defined as $\sum_{k=1}^{n}(Y_i - \bar{Y})^2$.

Thus, we can print the estimated values corresponding to the input values $\{x_1, \ldots, x_N\}$:

```r
fhat_2D = res$festimated
head(fhat_2D)

## [1] -0.001507484 -0.001594391 -0.001764006 -0.002086438 -0.002313565
## [6] -0.002730025
```

The value of the Residual Sum-of-square:

```r
res$rss

## [1] 1.910738
```

The value of the R-squared:

```r
res$rsq

## [1] 0.9988952
```

We can get the set of estimated knots for each dimension $\hat{t}_1$ and $\hat{t}_2$:

```r
knots.set = res$Selected.knots
print('For the first dimension: ')

## [1] "For the first dimension:"
print(knots.set[[1]])

## [1] 0.255 0.540

print('For the second dimension: ')

## [1] "For the second dimension:"
print(knots.set[[2]])

## [1] 0.650 0.655
```

As for $f_1$, we can visualize the corresponding estimation of $f_2$:
The red surface describes the true underlying function $f_2$ to estimate and the green surface corresponds to the estimation with GLOBER.

References
