Package 'localModel'

April 14, 2019

Title LIME-Based Explanations with Interpretable Inputs Based on Ceteris Paribus Profiles

Version 0.3.11

Author Mateusz Staniak [aut, cre],
Przemyslaw Biecek [aut],
Krystian Igras [ctb],
Alicja Gosiewska [ctb]

Maintainer Mateusz Staniak <m.staniak@mini.pw.edu.pl>

Description Local explanations of machine learning models describe, how features contributed to a single prediction.
This package implements an explanation method based on LIME
(Local Interpretable Model-agnostic Explanations,
see Tulio Ribeiro, Singh, Guestrin (2016) <doi:10.1145/2939672.2939778>) in which interpretable
inputs are created based on local rather than global behaviour of each original feature.

URL https://github.com/ModelOriented/localModel

BugReports https://github.com/ModelOriented/localModel/issues

Depends R (>= 3.5)

License GPL

Encoding UTF-8

LazyData true

Imports glmnet, ggplot2, partykit, ingredients

RoxygenNote 6.1.0

Suggests covr, knitr, rmarkdown, randomForest, DALEX, testthat

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2019-04-14 11:02:43 UTC

R topics documented:

`gaussian_kernel` *LIME kernel from the original article with sigma = 1.*

Description

Since only binary features are used, the weight associated with an observation is simply exp(-{number of features that were changed compared to the original observation}). Kernels are meant to be used as an argument to individual_surrogate_model function. Other custom functions can be used. Such functions take two vectors and return a single number.

Usage

```
gaussian_kernel(explained_instance, simulated_instance)
```

Arguments

`explained_instance`
: explained instance

`simulated_instance`
: new observation

Value

numeric

Examples

```
library(DALEX)
library(randomForest)
library(localModel)
data('apartments')
mrf <- randomForest(m2.price ~., data = apartments, ntree = 50)
explainer <- explain(model = mrf,
                     data = apartments[, -1])
model_lok <- individual_surrogate_model(explainer, apartments[5, -1],
                                        size = 500, seed = 17,
                                        kernel = gaussian_kernel)
# In this case each simulated observation has weight
# that is small when the distance from original observation is large,
```

```
# so closer observation have more weight.
model_lok
plot(model_lok)
```

`identity_kernel` *LIME kernel that treats all observations as equally similar to the observation of interest.*

Description

Kernels are meant to be used as an argument to individual_surrogate_model function. Other custom functions can be used. Such functions take two vectors and return a single number.

Usage

```
identity_kernel(explained_instance, simulated_instance)
```

Arguments

`explained_instance`	explained instance
`simulated_instance`	new observation

Value

numeric

Examples

```
library(DALEX)
library(randomForest)
library(localModel)
data('apartments')
mrf <- randomForest(m2.price ~., data = apartments, ntree = 50)
explainer <- explain(model = mrf,
                     data = apartments[, -1])
model_lok <- individual_surrogate_model(explainer, apartments[5, -1],
                                        size = 500, seed = 17,
                                        kernel = identity_kernel)
# In this case each simulated observation has equal weight
# when explanation model (LASSO) is fitted.
model_lok
plot(model_lok)
```

individual_surrogate_model *LIME-like explanations based on Ceteris Paribus curves*

Description

This function fits a LIME-type explanation of a single prediction. Interpretable binary features that describe the local impact of features on the prediction are created based on Ceteris Paribus Profiles. Thend, a new dataset of similar observations is created and black box model predictions (scores in case of classification) are calculated for this dataset and LASSO regression model is fitted to them. This way, explanations are simplified and include only the most important features. More details about the methodology can be found in the vignettes.

Usage

```
individual_surrogate_model(x, new_observation, size, seed = NULL,
  kernel = identity_kernel, sampling = "uniform", grid_points = 101)
```

Arguments

x	an explainer created with the function DALEX::explain().
new_observation	an observation to be explained. Columns in should correspond to columns in the data argument to x.
size	number of similar observation to be sampled.
seed	If not NULL, seed will be set to this value for reproducibility.
kernel	Kernel function which will be used to weight simulated observations.
sampling	Parameter that controls sampling while creating new observations.
grid_points	Number of points to use while calculating Ceteris Paribus profiles.

Value

data.frame of class local_surrogate_explainer

Examples

```
# Example based on apartments data from DALEX package.
library(DALEX)
library(randomForest)
library(localModel)
data('apartments')
mrf <- randomForest(m2.price ~., data = apartments, ntree = 50)
explainer <- explain(model = mrf,
                     data = apartments[, -1])
model_lok <- individual_surrogate_model(explainer, apartments[5, -1],
                                        size = 500, seed = 17)
```

```
model_lok
plot(model_lok)
```

`localModel` *localModel: LIME-like explanations with interpretable features based on Ceteris Paribus profiles*

Description

This package implements LIME-like explanation method (see Tulio Ribeiro, Singh, Guestrin (2016) <doi:10.1145/2939672.2939778>) in which interpretable inputs are created based on local rather than global behaviour of each original feature.#'

Important functions

`individual_surrogate_model` generates an explanation for a single prediction with interpretable features based on Ceteris Paribus profiles. `plot.local_surrogate_explainer` plots the explanation.

`plot.local_surrogate_explainer` *Generic plot function for local surrogate explainers*

Description

Generic plot function for local surrogate explainers

Usage

```
## S3 method for class 'local_surrogate_explainer'
plot(x, ..., geom = "point")
```

Arguments

`x`	object of class local_surrogate_explainer
`...`	other objects of class local_surrogate_explainer. If provided, models will be plotted in rows, response levels in columns.
`geom`	If "point", lines with points at the end will be plotted, if "bar", bars will be plotted and if "arrow", arrows.

Examples

```
# Example based on apartments data from DALEX package.
library(DALEX)
library(randomForest)
library(localModel)
data('apartments')
mrf <- randomForest(m2.price ~., data = apartments, ntree = 50)
explainer <- explain(model = mrf,
                     data = apartments[, -1])
model_lok <- individual_surrogate_model(explainer, apartments[5, -1],
                                        size = 500, seed = 17)
model_lok
plot(model_lok)
```

`print.local_surrogate_explainer` *Generic print function for local surrogate explainers*

Description

Generic print function for local surrogate explainers

Usage

```
## S3 method for class 'local_surrogate_explainer'
print(x, ...)
```

Arguments

`x`	object of class local_surrogate_explainer
`...`	currently ignored

Examples

```
# Example based on apartments data from DALEX package.
library(DALEX)
library(randomForest)
library(localModel)
data('apartments')
mrf <- randomForest(m2.price ~., data = apartments, ntree = 50)
explainer <- explain(model = mrf,
                     data = apartments[, -1])
model_lok <- individual_surrogate_model(explainer, apartments[5, -1],
                                        size = 500, seed = 17)
plot(model_lok)
model_lok
```

Index