Package ‘localModel’

September 14, 2021

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

Version 0.5

Maintainer  Przemyslaw Biecek <przemyslaw.biecek@gmail.com>

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

Imports  glmnet, DALEX, ggplot2, partykit, ingredients

RoxygenNote  7.1.1

Suggests  covr, knitr, rmarkdown, randomForest, testthat

VignetteBuilder  knitr

NeedsCompilation  no

Author  Przemyslaw Biecek [aut, cre], Mateusz Śniaki [aut], Krystian Igras [ctb], Alicja Gosiewska [ctb], Harel Lustiger [ctb], Willy Tadema [ctb]

Repository  CRAN

Date/Publication  2021-09-14 16:50:01 UTC
Description
Since only binary features are used, the weight associated with an observation is simply \( \exp\left(-\frac{\text{number of features that were changed compared to the original observation}}{2}\right) \). 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
```r
gaussian_kernel(explained_instance, simulated_instance)
```

Arguments
- `explained_instance`: explained instance
- `simulated_instance`: new observation

Value
numeric

Examples
```r
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)
```
identity_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. Then, 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

```r
individual_surrogate_model(
  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 = NULL,  # If not NULL, seed will be set to this value for reproducibility.
  kernel = identity_kernel,  # Kernel function which will be used to weight simulated observations.
  sampling = "uniform",  # Parameter that controls sampling while creating new observations.
  ...  # Additional arguments that will be passed to ingredients::ceteris_paribus.
)
```

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.
- `...` Additional arguments that will be passed to ingredients::ceteris_paribus.

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 explana-
tion.

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 = "bar")
plot_interpretable_feature

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)

plot_interpretable_feature

Plot Ceteris Paribus Profile and discretization

Description

Plot Ceteris Paribus Profile and discretization

Usage

plot_interpretable_feature(x, variable)

Arguments

x local_surrogate_explainer object

variable chr, name of the variable to be plotted

Value

ggplot2 object
print.local_surrogate_explainer

**Generic print function for local surrogate explainers**

### Description

Generic print function for local surrogate explainers

### Usage

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

### Arguments

- **x**
  - object of class local_surrogate_explainer
- **...**
  - currently ignored

### Examples

```r
# 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

gaussian_kernel. 2

identity_kernel. 3

individual_surrogate_model. 4, 5

localModel. 5

plot.local_surrogate_explainer. 5, 5

plot_interpretable_feature. 6

print.local_surrogate_explainer. 7