Package 'outForest'

January 7, 2021

Type Package

Title Multivariate Outlier Detection and Replacement

Version 0.1.1

Date 2021-01-06

Maintainer Michael Mayer <mayermichael79@gmail.com>

Description Provides a random forest based implementation of the method described in Chapter 7.1.2 (Regression model based anomaly detection) of Chandola et al. (2009) <doi:10.1145/1541880.1541882>. It works as follows: Each numeric variable is regressed onto all other variables by a random forest. If the scaled absolute difference between observed value and out-of-bag prediction of the corresponding random forest is suspiciously large, then a value is considered an outlier. The package offers different options to replace such outliers, e.g. by realistic values found via predictive mean matching. Once the method is trained on a reference data, it can be applied to new data.

License GPL (>= 2)

URL https://github.com/mayer79/outForest

BugReports https://github.com/mayer79/outForest/issues

Depends R (>= 3.5.0)

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports stats, graphics, FNN, ranger, missRanger (>= 2.1.0)

Suggests dplyr, knitr, rmarkdown

NeedsCompilation no

Author Michael Mayer [aut, cre]

Repository CRAN

Date/Publication 2021-01-07 02:50:02 UTC

R topics documented:

`Data` *Extracts Data*

Description

Extracts data with optionally replaced outliers from object of class 'outForest'.

Usage

```
Data(object, ...)

## Default S3 method:
Data(object, ...)

## S3 method for class 'outForest'
Data(object, ...)
```

Arguments

`object`	An object of class 'outForest'.
`...`	Arguments passed from or to other methods.

Value

A `data.frame`.

Methods (by class)

- `default`: Default method not implemented yet.
- `outForest`: Extract data from outForest object.

Examples

```
x <- outForest(iris)
head(Data(x))
```

`generateOutliers` *Adds Outliers to a Vector, Matrix or Data Frame*

Description

Takes a vector, matrix or data frame and replaces some numeric values by outliers.

Usage

```
generateOutliers(x, p = 0.05, sd_factor = 5, seed = NULL)
```

Arguments

`x`	A vector, matrix or `data.frame`.
`p`	Proportion of outliers to add to `x`. In case `x` is a `data.frame`, `p` can also be a vector of probabilities per column or a named vector (see examples).
`sd_factor`	Each outlier is generated by shifting the original value by a realization of a normal random variable with `sd_factor` times the original sample standard deviation.
`seed`	An integer seed.

Value

`x` with outliers.

See Also

`outForest`.

Examples

```
generateOutliers(1:10, seed = 334, p = 0.3)
generateOutliers(cbind(1:10, 10:1), p = 0.2)
head(generateOutliers(iris))
head(generateOutliers(iris, p = 0.2))
head(generateOutliers(iris, p = c(0, 0, 0.5, 0.5, 0.5)))
head(generateOutliers(iris, p = c(Sepal.Length = 0.2)))
```

is.outForest *Type Check*

Description

Checks if an object inherits class 'outForest'.

Usage

```
is.outForest(x)
```

Arguments

x	Any object.

Value

A logical vector of length one.

Examples

```
a <- outForest(iris)
is.outForest(a)
is.outForest("a")
```

outForest *Multivariate Outlier Detection and Replacement by Random Forest Predictions*

Description

This function provides a random forest based implementation of the method described in Chapter 7.1.2 ("Regression Model Based Anomaly detection") of Chandola et al. Each numeric variable to be checked for outliers is regressed onto all other variables using a random forest. If the scaled absolute difference between observed value and out-of-bag prediction is larger than some predefined threshold (default is 3), then a value is considered an outlier, see Details below. After identification of outliers, they can be replaced e.g. by predictive mean matching from the non-outliers. Since the random forest algorithm 'ranger' does not allow for missing values, any missing value is first being imputed by chained random forests. The method can be viewed as a multivariate extension of a basic univariate outlier detection method where a value is considered an outlier if it is more than e.g. three times the standard deviation away from its mean. In the multivariate case, instead of comparing a value with the overall mean, rather the difference to the conditional mean is considered. The 'outForest' function estimates this conditional mean by a random forest. If the method is trained on a reference data with option `allow_predictions`, it can be applied to new data.

Usage

```
outForest(
  data,
  formula = . ~ .,
  replace = c("pmm", "predictions", "NA", "no"),
  pmm.k = 3,
  threshold = 3,
  max_n_outliers = Inf,
  max_prop_outliers = 1,
  min.node.size = 40,
  allow_predictions = FALSE,
  impute_multivariate = TRUE,
  impute_multivariate_control = list(pmm.k = 3, num.trees = 50, maxiter = 3L),
  seed = NULL,
  verbose = 1,
  ...
)
```

Arguments

`data`	A `data.frame` to be assessed for numeric outliers.
`formula`	A two-sided formula specifying variables to be checked (left hand side) and variables used to check (right hand side). Defaults to . ~ ., i.e. use all variables to check all (numeric) variables.
`replace`	Should outliers be replaced by predicting mean matching on the OOB predictions ("pmm", the default), by OOB predictions ("predictions"), by `NA` ("NA"). Use "no" to keep outliers as they are.
`pmm.k`	For `replace = "pmm"`, how many nearest prediction neighbours (without outliers) be considered to sample observed values from?
`threshold`	Threshold above which an outlier score is considered an outlier. The default is 3.
`max_n_outliers`	Maximal number of outliers to identify. Will be used in combination with `threshold` and `max_prop_outliers`.
`max_prop_outliers`	Maximal relative count of outliers. Will be used in combination with `threshold` and `max_n_outliers`.
`min.node.size`	Minimal node size of the random forests. With 40, the value is relatively high. This reduces the impact of outliers.
`allow_predictions`	Should the resulting outForest be used on new data? Default is `FALSE` as fitted random forests can be huge.
`impute_multivariate`	If `TRUE` (default), missing values are imputed by `missRanger::missRanger`. Otherwise, by univariate sampling.
`impute_multivariate_control`	Parameters passed to `missRanger::missRanger` if data contains missing values.

seed	Integer random seed.
verbose	Controls how much outliers is printed to screen. 0 to print nothing, 1 prints information.
...	Arguments passed to ranger. If the data set is large, use less trees (e.g. num.trees = 20) and/or a low value of mtry.

Details

The outlier score of the i-th value x_ij of the j-th variable is defined as s_ij = (x_ij - pred_ij) / rmse_j, where pred_ij is the corresponding out-of-bag prediction of the j-th random forest and rmse_j its RMSE. If |s_ij| > L with threshold L, then x_ij is considered an outlier. For large data sets, just by chance, many values can surpass the default threshold of 3. To reduce the number of outliers, the threshold can be increased. Alternatively, the number of outliers can be limited by the two arguments max_n_outliers and max_prop_outliers. E.g. if at most ten outliers are to be identified, set max_n_outliers = 10.

Value

An object of type 'outForest' and a list with the following elements.

- Data: Original data set in unchanged row order but optionally with outliers replaced. Can be extracted with the Data function.
- outliers: Compact representation of outliers, for details see the outliers function used to extract them.
- n_outliers: Number of outliers per v.
- is_outlier: Logical matrix with outlier status. NULL if allow_predictions = FALSE.
- predData: data.frame with OOB predictions. NULL if allow_predictions = FALSE.
- allow_predictions: Same as allow_predictions.
- v: Variables checked.
- threshold: The threshold used.
- rmse: Named vector of RMSE of the random forests. Used for scaling the difference between observed values and predicted.
- forests: Named list of fitted random forests. NULL if allow_predictions = FALSE.
- used_to_check: Variables used for checking v.
- mu: Named vector of sample means of the original v (incl. outliers).

References

1. Chandola V., Banerjee A., and Kumar V. (2009). Anomaly detection: A survey. ACM Comput. Surv. 41, 3, Article 15 <dx.doi.org/10.1145/1541880.1541882>.
2. Wright, M. N. & Ziegler, A. (2016). ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R. Journal of Statistical Software, in press. <arxiv.org/abs/1508.04409>.

See Also

outliers, Data, plot.outForest, summary.outForest, predict.outForest.

Examples

```
head(irisWithOut <- generateOutliers(iris, seed = 345))
(out <- outForest(irisWithOut))
outliers(out)
head(Data(out))
plot(out)
plot(out, what = "scores")
```

`outliers` *Extracts Outliers*

Description

Extracts outliers from object of class 'outForest'. The outliers are sorted by their absolute score in descending fashion.

Usage

```
outliers(object, ...)

## Default S3 method:
outliers(object, ...)

## S3 method for class 'outForest'
outliers(object, ...)
```

Arguments

`object`	An object of class 'outForest'.
`...`	Arguments passed from or to other methods.

Value

A `data.frame` with one row per outlier. The columns are as follows:

- `row`, `col`: Row and column in original data with outlier.
- `observed`: Observed value.
- `predicted`: Predicted value.
- `rmse`: Scaling factor used to normalize the difference between observed and predicted.
- `score`: Outlier score defined as (observed-predicted)/rmse.
- `threshold`: Threshold above which an outlier score counts as outlier.
- `replacement`: Value used to replace observed value.

Methods (by class)

- `default`: Default method not implemented yet.
- `outForest`: Extract outliers from outForest object.

Examples

```
x <- outForest(iris)
outliers(x)
```

`plot.outForest` *Plot for outForest*

Description

This function can plot aspects of an 'outForest' object. For `what = "counts"`, the number of outliers per variable is visualized as a barplot. For `what = "scores"`, outlier scores (i.e. the scaled difference between predicted and observed value) are shown as scatter plot per variable.

Usage

```
## S3 method for class 'outForest'
plot(x, what = c("counts", "scores"), ...)
```

Arguments

`x`	An object of class `outForest`.
`what`	What should be plotted? One of "counts" (the default) or "scores".
`...`	Further arguments passed to `graphics::barplot` or `graphics::stripchart`.

Value

An object of class `ggplot2`.

Examples

```
irisWithOutliers <- generateOutliers(iris, seed = 345)
x <- outForest(irisWithOutliers, verbose = 0)
plot(x)
plot(x, what = "scores")
```

predict.outForest *Out-of-Sample Application*

Description

Identify outliers in new data set based on previously fitted 'outForest' object. The result of `predict` is again an object of type 'outForest'. All its methods can be applied to it.

Usage

```
## S3 method for class 'outForest'
predict(
  object,
  newdata,
  replace = c("pmm", "predictions", "NA", "no"),
  pmm.k = 3,
  threshold = object$threshold,
  max_n_outliers = Inf,
  max_prop_outliers = 1,
  seed = NULL,
  ...
)
```

Arguments

`object`	An object of class "outForest".
`newdata`	A new `data.frame` to be assessed for numeric outliers.
`replace`	Should outliers be replaced by predicting mean matching (from the original non-outliers) on the predictions ("pmm", the default), by predictions ("predictions"), by `NA` ("NA"). Use "no" to keep outliers as they are.
`pmm.k`	For `replace = "pmm"`, how many nearest prediction neighbours (from the original non-outliers) be considered to sample observed values from?
`threshold`	Threshold above which an outlier score is considered an outlier.
`max_n_outliers`	Maximal number of outliers to identify. Will be used in combination with `threshold` and `max_prop_outliers`.
`max_prop_outliers`	Maximal relative count of outliers. Will be used in combination with `threshold` and `max_n_outliers`.
`seed`	Integer random seed.
`...`	Further arguments passed from other methods.

Value

An object of type `outForest`.

See Also

outForest, outliers, Data.

Examples

```
(out <- outForest(iris, allow_predictions = TRUE))
iris1 <- iris[1, ]
iris1$Sepal.Length <- -1
pred <- predict(out, newdata = iris1)
outliers(pred)
Data(pred)
plot(pred)
plot(pred, what = "scores")
```

---

print.outForest *Prints outForest*

Description

Print method for an object of class outForest.

Usage

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

Arguments

x	A on object of class outForest.
...	Further arguments passed from other methods.

Value

Invisibly, the input is returned.

Examples

```
x <- outForest(iris)
x
```

process_scores *Process Scores*

Description

Internal function used to process scores and replace outliers.

Usage

```
process_scores(
  data,
  scores,
  predData,
  v,
  rmse,
  replace,
  pmm.k,
  threshold,
  max_n_outliers,
  max_prop_outliers,
  allow_predictions,
  obj = NULL
)
```

Arguments

data	Data set.
scores	Matrix with outlier scores.
predData	Prediction data.frame.
v	Variables checked.
rmse	rmse.
replace	replace.
pmm.k	pmm.k.
threshold	threshold.
max_n_outliers	max_n_outliers.
max_prop_outliers	max_prop_outliers.
allow_predictions	allow_predictions.
obj	outForest object.

Value

A list.

summary.outForest *Summarizes outForest*

Description

Summary method for an object of class outForest. Besides the number of outliers per variables, it shows the worst outliers.

Usage

```
## S3 method for class 'outForest'
summary(object, ...)
```

Arguments

object	A on object of class outForest.
...	Further arguments passed from other methods.

Value

A list of summary statistics.

Examples

```
out <- outForest(iris, seed = 34, verbose = 0)
summary(out)
```

Index