Function ‘SECOHDA’

October 1st, 2020

Type R Function (two versions, one for IPP and one for HMDH)

Title SECOHDA – SECODA based detection of high-density anomalies

Version 0.1

Author Ralph Foorthuis

Maintainer Ralph Foorthuis <tungusqa@hotmail.com>

Description SECOHDA essentially consists of two SECOHDA algorithms for the detection of high-density anomalies, namely IPP (Iterative Partial Push) or HMDH (Harmonic Mean Detection of HDAs). IPP and HMDH are algorithmic frameworks, using an underlying general-purpose anomaly detection algorithm (Foorthuis 2020). SECOHDA uses SECODA, a general-purpose unsupervised non-parametric anomaly detection algorithm for datasets containing continuous and/or categorical attributes (see Foorthuis 2017).

Imports SECODA, data.table

Encoding UTF-8

SECOHDA (IPP) *Segmentation- and Combination-Based detection of High-Density Anomalies*

Description

The IPP framework for the detection of high-density anomalies using SECODA as underlying algorithm. Contained in file “SECOHDA_AlgorithmIterativePP.R”.

Usage

```
SECOHDA(
  datset,
  QuantileDenominator = 100,
  QuantileFilterBoost = -9999,
  TestMode=TRUE,
  MinimumNumberOfSECODAIterations=7,
  StartSECODAHeuristicsAfterIteration=10
)
```

Arguments

`datset`	The dataset that is being analyzed for anomalies. Datset should be a data.frame. SECODA treats numeric and categorical data differently. Before running SECOHDA() make sure that the data types are declared correctly. Numeric data should be 'integer' or 'numeric', whereas categorical data should be 'factor', 'logical' or 'character'.
`QuantileDenominator`	The QuantileDenominator determines the degree of detail during the analysis. The higher the parameter, the more detailed the analysis (and the slower the performance). The QuantileDenominator is typically 100 or 1000.
`QuantileFilterBoost`	The QuantileFilterBoost represents the (in)sensitivity. It is the percentage of numerically isolated anomalies that are additionally filtered away from the general anomalies. The higher the QuantileFilterBoost, the less risk of isolated anomalies being labelled as most extreme anomalies (i.e. less false positives amongst the lowest scores), but the higher the risk of true high-density anomalies being missed (i.e. more false negatives). A value of -9999 means this setting is determined automatically.
`TestMode`	The mode for returning information regarding the analysis process. • "TRUE" for showing messages. • "FALSE" for not showing messages.
`MinimumNumberOfSECODA Iterations`	The minimum number of iterations performed by SECODA. The algorithm will conduct at least this number of iterations, even if it has converged. This setting can be increased to make the results more precise when running time is not an issue. Standard value is 7, but can be set to a lower value in experimental situations (SECODA will then decide itself if fewer iterations will suffice).
`StartSECODAHeuristics AfterIteration`	The iteration after which several heuristics will be applied by SECODA. These heuristics speed up the process, but make the results somewhat less precise. In HighDimMode "IN" it is recommended to set this argument as low as possible, depending mainly on the precision with which the user wants to discretize the continuous variables. If 5 bins (intervals) is sufficient, the StartHeuristicsAfterIteration argument can be set to 4, if 9 bins is sufficient, StartHeuristicsAfterIteration can be set to 8, et cetera. Also see the DSAA 2017 paper by Foorthuis for information on the pruning heuristic that is triggered when the number of iterations set by this argument is reached.

Value

SECOHDA returns a data frame containing the ID and an high-density anomaly score for all cases in the original input dataset 'datset'. Low scores represent anomalous cases. The Ano_ID is the row number of the case in the original 'datset'.

Author(s)

Ralph Foorthuis

References

Foorthuis, R.M. (2020). *Algorithmic Frameworks for the Detection of High-Density Anomalies*. Accepted for presentation at IEEE SSCI CIDM 2020 (Symposium on Computational Intelligence in Data Mining), December 2020, Canberra Australia.

Foorthuis, R.M. (2017). *SECODA: Segmentation- and Combination-Based Detection of Anomalies*. In: Proceedings of the 4th IEEE International Conference on Data Science and Advanced Analytics (DSAA 2017), Tokyo, Japan.

SECODA and SECOHDA example data files and R code: http://www.foorthuis.nl

Examples

```
## Not run:
SECOHDA(Data1)

SECOHDA(Data1, StartSECODAHeuristicsAfterIteration = 9999, TestMode = TRUE)
# Make sure SECODA heuristics are not applied, and messages are shown on the
screen.

SECOHDA(Data1, StartSECODAHeuristicsAfterIteration = 9999, QuantileFilterBoost = 4)
# Make sure SECODA heuristics are not applied, and QuantileFilterBoost is set at 4.

# See www.foorthuis.nl for example data files and R code.

## End(Not run)
```

Description

The HMDH framework for the detection of high-density anomalies using SECODA as underlying algorithm. Contained in file “SECOHDA_AlgorithmHarmonicMean.R”.

Usage

```
SECOHDA(
  datset,
  WeightCorrection="SDEN",
  TestMode=TRUE,
  MinimumNumberOfSECODAIterations=7,
  StartSECODAHeuristicsAfterIteration=10
)
```

Arguments

`datset`	The dataset that is being analyzed for anomalies. Datset should be a data.frame. SECODA treats numeric and categorical data differently. Before running SECOHDA() make sure that the data types are declared correctly. Numeric data should be 'integer' or 'numeric', whereas categorical data should be 'factor', 'logical' or 'character'.
`WeightCorrection`	The method used for the correction of *ads/aas* weights (see Foorthuis 2020). • "None": no correction. • "SDEN": single-class density. • "SSE": single Shannon entropy.
`TestMode`	The mode for returning information regarding the analysis process. • "TRUE" for showing messages. • "FALSE" for not showing messages.
`MinimumNumberOfSECODA Iterations`	The minimum number of iterations performed by SECODA. The algorithm will conduct at least this number of iterations, even if it has converged. This setting can be increased to make the results more precise when running time is not an issue. Standard value is 7, but can be set to a lower value in experimental situations (SECODA will then decide itself if fewer iterations will suffice).
`StartSECODAHeuristics AfterIteration`	The iteration after which several heuristics will be applied by SECODA. These heuristics speed up the process, but make the results somewhat less precise. In HighDimMode "IN" it is recommended to set this argument as low as possible, depending mainly on the precision with which the user wants to discretize the continuous variables. If 5 bins (intervals) is sufficient, the StartHeuristicsAfterIteration argument can be set to 4, if 9 bins is sufficient, StartHeuristicsAfterIteration can be set to 8, et cetera. Also see the DSAA 2017 paper by Foorthuis for information on the pruning heuristic that is triggered when the number of iterations set by this argument is reached.

Value

SECOHDA returns a data frame containing the ID and an high-density anomaly score for all cases in the original input dataset 'datset'. Low scores represent anomalous cases. The Ano_ID is the row number of the case in the original 'datset'.

Author(s)

Ralph Foorthuis

References

Foorthuis, R.M. (2020). *Algorithmic Frameworks for the Detection of High-Density Anomalies*. Accepted for presentation at IEEE SSCI CIDM 2020 (Symposium on Computational Intelligence in Data Mining), December 2020, Canberra Australia.

Foorthuis, R.M. (2017). *SECODA: Segmentation- and Combination-Based Detection of Anomalies*. In: Proceedings of the 4th IEEE International Conference on Data Science and Advanced Analytics (DSAA 2017), Tokyo, Japan.

SECODA and SECOHDA example data files and R code: http://www.foorthuis.nl

Examples

```
## Not run:
SECOHDA(DataSet1)

SECOHDA(DataSet1, WeightCorrection = "None", TestMode = TRUE) # Do not use weight
correction, and show messages.

SECOHDA(DataSet1, WeightCorrection = "SDEN", StartSECODAHeuristicsAfterIteration =
9999, TestMode = TRUE) # Make sure SDEN is used, SECODA heuristics are not applied,
and messages are shown on the screen.

# See www.foorthuis.nl for example data files and R code.

## End(Not run)
```