Package ‘SECODA’

February 28, 2020

Type Package

Title SECODA anomaly detection

Version 0.5.4

Author Ralph Foorthuis

Maintainer Ralph Foorthuis <tungusqa@hotmail.com>

Description SECODA is a novel general-purpose unsupervised non-parametric anomaly detection algorithm for datasets containing continuous and/or categorical attributes. The method, in its standard mode, is guaranteed to identify cases with unique or sparse combinations of attribute values.

SECODA identifies different types of anomalies (see Foorthuis 2018 for more information about anomaly types):

- **I. Extreme value anomaly**: A case with an extremely high or low (or otherwise rare) value. A case can be an anomaly on one individual attribute, so *extreme value anomalies* do not depend on relationships between attributes.
- **II. Rare class anomaly**: A case with a rare class value on one or multiple categorical attributes. A case can be an anomaly on one individual attribute, so *sparse class anomalies* do not depend on relationships between attributes.
- **III. Simple mixed data anomaly**: A case that is both a Type I and Type II anomaly, i.e. with at least one extreme value and one rare class. This requires deviant values for at least two attributes, each anomalous in its own right. These can thus be analyzed separately; analyzing the attributes jointly is unnecessary because the case is not anomalous in terms of a combination of values.
- **IV. Multidimensional numerical anomaly**: A case that does not conform to the general pattern when multiple numerical attributes are taken into account jointly, but does not have extreme values for any of its individual numerical attributes.
- **V. Multidimensional rare class anomaly**: A case with a rare combination of class values. A minimum of two substantive categorical attributes needs to be analyzed jointly to discover a multidimensional rare class anomaly (at least in datasets with independent data points).
- **VI. Multidimensional mixed data anomaly**: A case with a categorical value or a combination of categorical values that in itself is not rare in the dataset as a whole, but is only rare in its neighborhood (numerical area).

The algorithm uses the histogram-based approach to assess the density. The concatenation trick – which combines discretized continuous attributes and categorical attributes into a new variable – is used to determine the joint density distribution. In combination with recursive discretization this captures complex relationships between attributes and avoids discretization error. A pruning heuristic as well as exponentially increasing weights and arity are employed to speed up the analysis.

The user is advised to first try the standard HighDimMode setting ("NO") for low-dimensional datasets. If the algorithm returns the warning that the given fraction of anomalies was already reached in first iteration, this is probably due to the curse of dimensionality. The user should consequently re-run the analysis with a HighDimMode setting for higher-dimensional datasets (first "CA", then "IN"). Note that the "IN" setting does not guarantee that unique attribute value combinations are identified as anomalies. See below for more information on the HighDimMode options.

Imports data.table

Encoding UTF-8

SECODA *Segmentation- and Combination-Based Detection of Anomalies (SECODA) algorithm*

Description

The SECODA algorithm for Segmentation- and Combination-Based Detection of Anomalies.

Usage

```
SECODA(
  datset,
  BinningMethod = "EW",
  HighDimMode = "NO",
  MinimumNumberOfIterations = 7,
  MaximumNumberOfIterations = 99999,
  StartHeuristicsAfterIteration = 10,
  FractionOfCasesToRetain = 0.2,
  InitialArity = 2,
  TestMode = "Normal"
)
```

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 SECODA() 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'.
`BinningMethod`	The method used for unsupervised discretization. • "EW" for equiwidth (equal interval) binning. • "ED" for equidepth (equal frequency) binning.
`HighDimMode`	The approach to deal with low- or high-dimensionality. This setting may be changed when SECODA displays a warning message that there may be too many variables in the set, which is triggered if the algorithm already reached the convergence criterion after 1 iteration. • "NO" is the *normal* (standard) mode. This mode is able to detect all four types of anomalies. It guarantees that all unique attribute value combinations are identified as anomalies in the most efficient way, but assumes there are not too many attributes (i.e. is vulnerable to the curse of dimensionality). • "CA" mode conducts the first iteration only with *categorical* attributes. This mode generally is able to analyze a larger number of attributes than NO (normal) mode and still guarantees that all unique attribute value combinations are identified as anomalies. However, it runs somewhat less efficient than NO mode. Also, the CA mode gives categorical attributes a somewhat higher weight (which can be compensated partially by increasing the MinimumNumberOfIterations if time performance is not an issue). The CA mode assumes there are sufficient continuous attributes (that can be ignored in the first iteration) and that the joint categorical attributes do not yield too many unique combinations (i.e. constellations) in the first iterations. As such, it is still vulnerable to the curse of dimensionality.

- "IN" mode combines the NO mode with a univariate analysis, which focuses on the *individual* variables separately. The IN mode is able to deal with a large amount of attributes, but is no longer guaranteed to identify unique attribute value combinations as the most extreme anomalies. There are two phases in any given iteration. The *univariate*phase analyzes the individual attributes without taking into account the relationships between them (i.e. combinations of attribute values are ignored). The *multivariate* phase within the iteration does take the relationships between variables into account. SECODA stops running the univariate phase when the number of iterations specified in the StartHeuristicsAfterIteration argument is reached, so the user can decrease this number to give relationships between variables more weight in the analysis. It is recommended to set this setting as low as possible (see StartHeuristicsAfterIteration).

`MinimumNumberOfIterations` The minimum number of iterations. 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).

`MaximumNumberOfIterations` The maximum number of iterations. The algorithm will conduct at most this number of iterations, even if it has not yet converged the regular way. This can speed up the analysis at the cost of precision. Furthermore, although the algorithm is designed to avoid infinite loops, it is still possible that a certain combination of settings results in an endless loop. The analysis can then be rerun with an acceptable value for MaximumNumberOfIterations.

`StartHeuristicsAfterIteration` The iteration after which several heuristics will be applied. 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.

`FractionOfCasesToRetain` The fraction of cases to retain while running the pruning heuristic. The fraction is set as a number between 0 and 1. The pruning will not discard more cases than 1-FractionOfCasesToRetain.

`InitialArity` The number of discretization intervals (bins) for the first iteration. Standard value is 2 intervals.

`TestMode` The mode for returning information regarding the analysis process.

- "Normal" for simply returning the anomaly scores.
- "FullReturn" for returning the anomaly scores and process log, but not displaying the full messages while running.
- "FullTest" for returning the anomaly scores and process log, and displaying the full messages while running.

Value

SECODA returns a data frame containing the ID and an anomaly score for all cases in the original input dataset 'datset'. Low scores represent anomalous cases. The ID is the row number of the case in the original 'datset'. If TestMode is FullReturn or FullTest the process log is also returned.

Author(s)

Ralph Foorthuis

References

Foorthuis, R.M. (2019). *All or In-cloud: How the Identification of Six Types of Anomalies is Affected by the Discretization Method*. In: Atzmueller M., Duivesteijn W. (eds) Artificial Intelligence. BNAIC 2018. Springer, Communications in Computer and Information Science, Vol. 1021, pp 25-42. DOI: 10.1007/978-3-030-31978-6_3.

Foorthuis, R.M. (2018). *A Typology of Data Anomalies*. In: Springer CCIS 854, Proceedings of the 17th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2018), Cádiz, Spain. DOI: 10.1007/978-3-319-91476-3_3.

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.

Foorthuis, R.M. (2017). *Anomaly Detection with SECODA*. Poster Presentation at the 4th IEEE International Conference on Data Science and Advanced Analytics (DSAA 2017), Tokyo, Japan.

Foorthuis, R.M. (2017). *The SECODA Algorithm for the Detection of Anomalies in Sets with Mixed Data*. Presentation.

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

Examples

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

SECODA(DataSet1, MinimumNumberOfIterations = 12, StartHeuristicsAfterIteration =
15) # Make sure at least 12 iterations are run, and, if needed, start heuristics
after 15 iterations.

SECODA(DataSet1, HighDimMode = "CA") # You can also use "IN", for dealing with
high-dimensional datasets.

SECODA(DataSet1, BinningMethod = "ED") # Use equidepth (equal frequency)
discretization instead of the standard equiwidth (equal interval) discretization.

SECODA(DataSet1, TestMode="FullTest") # See messages in the R console.

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

## End(Not run)
```