Package ‘subgroup.discovery’

August 1, 2017

Type Package

Title Subgroup Discovery and Bump Hunting

Version 0.2.0

Description Developed to assist in discovering interesting subgroups in high-dimensional data. The PRIM implementation is based on the 1998 paper ``Bump hunting in high-dimensional data'' by Jerome H. Friedman and Nicholas I. Fisher. <doi:10.1023/A:1008894516817> PRIM involves finding a set of ``rules'' which combined imply unusually large (or small) values of some other target variable. Specifically one tries to find a set of sub regions in which the target variable is substantially larger than overall mean. The objective of bump hunting in general is to find regions in the input (attribute/feature) space with relatively high (low) values for the target variable. The regions are described by simple rules of the type if: condition-1 and ... and condition-n then: estimated target value. Given the data (or a subset of the data), the goal is to produce a box B within which the target mean is as large as possible. There are many problems where finding such regions is of considerable practical interest. Often these are problems where a decision maker can in a sense choose or select the values of the input variables so as to optimize the value of the target variable. In bump hunting it is customary to follow a so-called covering strategy. This means that the same box construction (rule induction) algorithm is applied sequentially to subsets of the data.

Depends R (>= 2.10)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

URL `https://github.com/Jurian/subgroup.discovery`

BugReports `https://github.com/Jurian/subgroup.discovery/issues`

Date 2017-07-15

Suggests testthat

Author Jurian Baas [aut, cre, cph],
Ad Feelders [ctb]

Maintainer Jurian Baas `<jurian@jurianbaas.nl>`

R topics documented:

`ames` *Ames Housing data.*

Description

Data set contains information from the Ames Assessor Office used in computing assessed values for individual residential properties sold in Ames, IA from 2006 to 2010. Tab characters are used to separate variables in the data file. The data has 82 columns which include 23 nominal, 23 ordinal, 14 discrete, and 20 continuous variables (and 2 additional observation identifiers).

Usage

```
ames
```

Format

A data frame with 2930 rows and 82 variables:

Order (Discrete): Observation number

PID (Nominal): Parcel identification number - can be used with city web site for parcel review

MS.SubClass (Nominal): Identifies the type of dwelling involved in the sale

MS.Zoning (Nominal): Identifies the general zoning classification of the sale

Lot.Frontage (Continuous): Linear feet of street connected to property

Lot.Area (Continuous): Lot size in square feet

Street (Nominal): Type of road access to property

Alley (Nominal): Type of alley access to property

Lot.Shape (Ordinal): General shape of property

Land.Contour (Nominal): Flatness of the property

Utilities (Ordinal): Type of utilities available

Lot.Config (Nominal): Lot configuration

Land.Slope (Ordinal): Slope of property

Neighborhood (Nominal): Physical locations within Ames city limits (map available)

Condition.1 (Nominal): Proximity to various conditions

Condition.2 (Nominal): Proximity to various conditions (if more than one is present)

Bldg.Type (Nominal): Type of dwelling

House.Style (Nominal): Style of dwelling

Overall.Qual (Ordinal): Rates the overall material and finish of the house

Overall.Cond (Ordinal): Rates the overall condition of the house

Year.Built (Discrete): Original construction date

Year.Remod.Add (Discrete): Remodel date (same as construction date if no remodeling or additions)

Roof.Style (Nominal): Type of roof

Roof.Matl (Nominal): Roof material

Exterior.1st (Nominal): Exterior covering on house

Exterior.2nd (Nominal): Exterior covering on house (if more than one material)

Mas.Vnr.Type (Nominal): Masonry veneer type

Mas.Vnr.Area (Continuous): Masonry veneer area in square feet

Exter.Qual (Ordinal): Evaluates the quality of the material on the exterior

Exter.Cond (Ordinal): Evaluates the present condition of the material on the exterior

Foundation (Nominal): Type of foundation

Bsmt.Qual (Ordinal): Evaluates the height of the basement

Bsmt.Cond (Ordinal): Evaluates the general condition of the basement

Bsmt.Exposure (Ordinal): Refers to walkout or garden level walls

BsmtFin.Type.1 (Ordinal): Rating of basement finished area

BsmtFin.SF.1 (Continuous): Type 1 finished square feet

BsmtFin.Type.2 (Ordinal): Rating of basement finished area (if multiple types)

BsmtFin.SF.2 (Continuous): Type 2 finished square feet

Bsmt.Unf.SF (Continuous): Unfinished square feet of basement area

Total.Bsmt.SF (Continuous): Total square feet of basement area

Heating (Nominal): Type of heating

Heating.QC (Ordinal): Heating quality and condition

Central.Air (Nominal): Central air conditioning

Electrical (Ordinal): Electrical system

X1st.Flr.SF (Continuous): First Floor square feet

X2nd.Flr.SF (Continuous) : Second floor square feet

Low.Qual.Fin.SF (Continuous): Low quality finished square feet (all floors)

Gr.Liv.Area (Continuous): Above grade (ground) living area square feet

Bsmt.Full.Bath (Discrete): Basement full bathrooms

Bsmt.Half.Bath (Discrete): Basement half bathrooms

Full.Bath (Discrete): Full bathrooms above grade

Half.Bath (Discrete): Half baths above grade

Bedroom.AbvGr (Discrete): Bedrooms above grade (does NOT include basement bedrooms)

Kitchen.AbvGr (Discrete): Kitchens above grade

Kitchen.Qual (Ordinal): Kitchen quality

TotRms.AbvGrd (Discrete): Total rooms above grade (does not include bathrooms)

Functional (Ordinal): Home functionality (Assume typical unless deductions are warranted)

Fireplaces (Discrete): Number of fireplaces

Fireplace.Qu (Ordinal): Fireplace quality

Garage.Type (Nominal): Garage location

Garage.Yr.Blt (Discrete): Year garage was built

Garage.Finish (Ordinal) : Interior finish of the garage

Garage.Cars (Discrete): Size of garage in car capacity

Garage.Area (Continuous): Size of garage in square feet

Garage.Qual (Ordinal): Garage quality

Garage.Cond (Ordinal): Garage condition

Paved.Drive (Ordinal): Paved driveway

Wood.Deck.SF (Continuous): Wood deck area in square feet

Open.Porch.SF (Continuous): Open porch area in square feet

Enclosed.Porch (Continuous): Enclosed porch area in square feet

X3Ssn.Porch (Continuous): Three season porch area in square feet

Screen.Porch (Continuous): Screen porch area in square feet

Pool.Area (Continuous): Pool area in square feet

Pool.QC (Ordinal): Pool quality

Fence (Ordinal): Fence quality

Misc.Feature (Nominal): Miscellaneous feature not covered in other categories

Misc.Val (Continuous): $Value of miscellaneous feature

Mo.Sold (Discrete): Month Sold (MM)

Yr.Sold (Discrete): Year Sold (YYYY)

Sale.Type (Nominal): Type of sale

Sale.Condition (Nominal): Condition of sale

SalePrice (Continuous): Sale price

Details

Sources: Ames, Iowa Assessor Office

Source

https://ww2.amstat.org/publications/jse/v19n3/decock/datadocumentation.txt

---

`credit` *Credit scoring data.*

Description

A dataset containing the attributes of people who did or did not default on their loan.

Usage

```
credit
```

Format

A data frame with 10 rows and 6 variables:

age age of person

married is the person married or not, boolean

house is the person a homeowner or not, boolean

income income of person, in thousands

gender factor with levels male, female

class class variable (0 or 1)

Details

Toy example, useful for debugging purposes.

Source

http://www.cs.uu.nl/docs/vakken/adm/bump.pdf

pima *Pima Indians Diabetes Database.*

Description

Sources: (a) Original owners: National Institute of Diabetes and Digestive and Kidney Diseases (b) Donor of database: Vincent Sigillito <vgs@aplcen.apl.jhu.edu> Research Center, RMI Group Leader Applied Physics Laboratory The Johns Hopkins University Johns Hopkins Road Laurel, MD 20707 (301) 953-6231 (c) Date received: 9 May 1990

Usage

```
pima
```

Format

A data frame with 768 rows and 9 variables:

pregnant Number of times pregnant

glucose Plasma glucose concentration a 2 hours in an oral glucose tolerance test

bp Diastolic blood pressure (mm Hg)

skin_thickness Triceps skin fold thickness (mm)

insulin 2-Hour serum insulin (mu U/ml)

bmi Body mass index (weight in kg/(height in m)^2)

diabetes Diabetes pedigree function

age Age (years)

class Class variable (0 or 1)

Details

Past Usage: Smith, J.W., Everhart, J.E., Dickson, W.C., Knowler, W.C., & Johannes, R.S. (1988). Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In Proceedings of the Symposium on Computer Applications and Medical Care (pp. 261–265). IEEE Computer Society Press.

The diagnostic, binary-valued variable investigated is whether the patient shows signs of diabetes according to World Health Organization criteria (i.e., if the 2 hour post-load plasma glucose was at least 200 mg/dl at any survey examination or if found during routine medical care). The population lives near Phoenix, Arizona, USA.

Results: Their ADAP algorithm makes a real-valued prediction between 0 and 1. This was transformed into a binary decision using a cutoff of 0.448. Using 576 training instances, the sensitivity and specificity of their algorithm was 76

Relevant Information: Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

Source

https://archive.ics.uci.edu/ml/datasets/pima+indians+diabetes

`plot.prim.cover` *Plot PRIM cover result*

Description

Plot an S3 object of class prim.cover

Usage

```
## S3 method for class 'prim.cover'
plot(x, ...)
```

Arguments

`x`	An S3 object of class prim.cover
`...`	Optional arguments to pass on

Value

Nothing, this function is called for its side-effects

Author(s)

Jurian Baas

`plot.prim.diversify` *Plot PRIM diversify result*

Description

Plot an S3 object of class prim.diversify

Usage

```
## S3 method for class 'prim.diversify'
plot(x, ...)
```

Arguments

`x`	An S3 object of class prim.diversify
`...`	Optional arguments to pass on

Value

Nothing, this function is called for its side-effects

Author(s)

Jurian Baas

`plot.prim.peel` *Plot PRIM peel result*

Description

Plot an S3 object of class prim.peel

Usage

```
## S3 method for class 'prim.peel'
plot(x, ...)
```

Arguments

`x`	An S3 object of class prim.peel
`...`	Optional arguments to pass on

Value

Nothing, this function is called for its side-effects

Author(s)

Jurian Baas

`plot.prim.validate` *Plot PRIM test result*

Description

Plot an S3 object of class prim.validate

Usage

```
## S3 method for class 'prim.validate'
plot(x, ...)
```

Arguments

`x`	An S3 object of class prim.validate
`...`	Optional arguments to pass on

Value

Nothing, this function is called for its side-effects

Author(s)

Jurian Baas

predict.prim.cover *Predict method for PRIM Cover Fits*

Description

Predicted values based on the PRIM cover object

Usage

```
## S3 method for class 'prim.cover'
predict(object, newdata, ...)
```

Arguments

`object`	Object of class prim.cover
`newdata`	A data frame in which to look for variables with which to predict.
`...`	Further arguments passed to or from other methods

Value

Depends on the quality function used. In the case of base::mean, the mean of the target variable of the first matching box.

Author(s)

Jurian Baas

prim.box.optimal *Find the optimal box depending on the strategy*

Description

Finds the box with the highest quality or the box closest to the maximum quality minus 2 times the standard error

Usage

```
prim.box.optimal(prim.validate)
```

Arguments

`prim.validate`	An object of type "prim.validate"

Value

The index of the optimal box

Author(s)

Jurian Baas

`prim.candidates.find` *PRIM find split candidates*

Description

Find all box candidates for a given (sub)set

Usage

```
prim.candidates.find(X, y, peeling.quantile, min.support, max.peel,
  quality.function)
```

Arguments

`X`	Data frame with observations (may be a subset of original data)
`y`	Dependent variable, usually a numeric vector
`peeling.quantile`	Quantile to peel off
`min.support`	Minimal size of a box
`max.peel`	Maximal size of a peel
`quality.function`	Function to use to determine box quality

Details

This function goes through all columns of the dataset and tries to findbox candidates based on the quantile peeling.quantile and minimum support min.support. Note that the indexes returned are those that have to be removed in order to create the box!

Value

A list of potential boxes

Author(s)

Jurian Baas

`prim.cover` *PRIM covering strategy*

Description

In bump hunting it is customary to follow a so-called covering strategy. This means that the same box construction (rule induction) algorithm is applied sequentially to subsets of the data.

Usage

```
prim.cover(formula, data, X, y, peeling.quantile = 0.03, min.support = 0.05,
  max.peel = 0.1, train.fraction = 0.66, max.boxes = NA,
  quality.function = base::mean, plot = FALSE, minimize = FALSE,
  optimal.box = c("best", "2se"))
```

Arguments

`formula`	Formula with a response and terms
`data`	Data frame to find rules in
`X`	Optionally instead of using a formula: Data frame to find rules in
`y`	Optionally instead of using a formula: Response vector, usually of type numeric
`peeling.quantile`	Quantile to peel off for numerical variables
`min.support`	Minimal size of a box to be valid
`max.peel`	Maximal size of a peel, as a fraction. Defaults to 0.1
`train.fraction`	Train-test split fraction used in validation, defaults to 0.66
`max.boxes`	Maximum number of boxes, NA or leave out for no limit
`quality.function`	Function to use for determining set quality, defaults to mean
`plot`	Plot intermediate results, defaults to false
`minimize`	Should the quality be minimized? Same as setting the quality function to function(x)-quality.function(x). Defaults to FALSE
`optimal.box`	During validation, choose the box with the highest quality or a simpler box, two standard errors from the optimum

Value

An S3 object of class prim.cover

Author(s)

Jurian Baas

Examples

```
data(pima)
p.cov <- prim.cover(
    class ~ .,
    data = pima,
    peeling.quantile = 0.05,
    min.support = 0.1,
    plot = TRUE,
    optimal.box = "2se"
)


summary(p.cov)
plot(p.cov)
```

prim.diversify *PRIM diversify strategy*

Description

Provide a (hopefully) diverse number of box definitions

Usage

```
prim.diversify(formula, data, X, y, n, peeling.quantile = 0.03,
  min.support = 0.05, max.peel = 0.1, train.fraction = 0.66,
  quality.function = base::mean, plot = FALSE, parallel = TRUE,
  minimize = FALSE, optimal.box = c("best", "2se"))
```

Arguments

`formula` Formula with a response and terms

`data` Data frame to find rules in

`X` Optionally instead of using a formula: Data frame to find rules in

`y` Optionally instead of using a formula: Response vector, usually of type numeric

`n` Numer of attempts to run the PRIM algorithm

`peeling.quantile` Quantile to peel off for numerical variables

`min.support` Minimal size of a box to be valid

`max.peel` Maximal size of a peel, as a fraction. Defaults to 0.1

`train.fraction` Train-test split fraction used in validation, defaults to 0.66

`quality.function` Function to use for determining subset quality, defaults to mean

`plot` Plot intermediate results, defaults to false. Note that intermediate plotting is unavailable when running in parallel

`parallel` Compute each run in parallel, defaults to TRUE. This will use all but one core. Note that intermediate plotting is unavailable when running in parallel

`minimize` Should the quality be minimized? Same as setting the quality function to function(x)-quality.function(x). Defaults to FALSE

`optimal.box` During validation, choose the box with the highest quality or a simpler box, two standard errors from the optimum

Details

Because the final box depends on the data used, we re-run the PRIM peeling algorithm multiple times, each with a different random train/test split. Each run is independent from the others, so this algorithm is run in parallel by default.

Value

An S3 object of type prim.diversify

Author(s)

Jurian Baas

Examples

```
data(ames)
p.div <- prim.diversify(
    SalePrice ~ . - PID - Order,
    data = ames,
    n = 5,
    plot = TRUE,
    parallel = FALSE,
    optimal.box = "best"
)


summary(p.div)
plot(p.div)
```

prim.diversify.compare

Compare PRIM diversify results

Description

Compares all attempts of a PRIM diversify operation with each other

Usage

```
prim.diversify.compare(X, p.div)
```

Arguments

`X`	Data frame to do intersect and union operations
`p.div`	An S3 object of type "prim.diversify"

Details

Comparison is done by the following formula $\frac{|A \cap B|}{|A \cup B|}$

Value

A matrix with the comparisons laid out by position

Author(s)

Jurian Baas

prim.peel *Bump hunting using the Patient Rule Induction Method*

Description

Peeling function for bump hunting using the Patient Rule Induction Method (PRIM).

Usage

```
prim.peel(X, y, N, peeling.quantile, min.support, max.peel, quality.function)
```

Arguments

`X`	Data frame to find rules in
`y`	Response vector, usually of type numeric
`N`	Size of entire data set
`peeling.quantile`	Quantile to peel off for numerical variables
`min.support`	Minimal size of a box to be valid, as a fraction
`max.peel`	Maximal size of a peel, as fraction
`quality.function`	Which function to use to determine the quality of a box

Value

An S3 object of class prim.peel

Author(s)

Jurian Baas

prim.rule.condense *Condense multiple (redundant) rules*

Description

This function condenses the many (redundant) rules of an S3 object of class prim.peel or prim.validate to a single rule.

Usage

```
prim.rule.condense(prim.object)
```

Arguments

`prim.object`	An S3 object of class prim.validate

Details

This function condenses the many (redundant) rules of an S3 object of class prim.peel or prim.validate to a single rule.

Value

The condensed rule as a single string

Author(s)

Jurian Baas

`prim.rule.match` *Create box matching index*

Description

Generate a logical vector in which elements are true iff applying the rule to a record evaluates to true.

Usage

```
prim.rule.match(prim.object, X)
```

Arguments

`prim.object`	An S3 object of class prim.peel or prim.validate result
`X`	A data frame with at least those columns that were used in creating the prim S3 object

Value

A logical index of matching records

Author(s)

Jurian Baas

prim.rule.operations *Intersection of multiple rules*

Description

This function applies the rules given by the parameters to a dataset and calculates the intersection, i.e. those observations where all rules evaluate to TRUE are returned as TRUE.

Usage

```
prim.rule.operations(X, prim.objects, operation = c("union", "intersect"))
```

Arguments

`X`	Data frame to apply rules to
`prim.objects`	A list of objects of class "prim.peel" and/or "prim.validate"
`operation`	One of "union", "intersect"

Value

Logical vector, true iff all rules evaluate to TRUE for a certain observation

Author(s)

Jurian Baas

prim.validate *Bump hunting using the Patient Rule Induction Method*

Description

Validate the results taken from the PRIM peeling process

Usage

```
prim.validate(peel.result, X, y, optimal.box)
```

Arguments

`peel.result`	An S3 object of class prim.peel
`X`	A data frame with at least those columns that were used in creating the prim.peel S3 object
`y`	Response vector, usually of type numeric
`optimal.box`	Choose the box with the highest quality or a simpler box, two standard errors from the optimum

Details

This function takes the result of the prim peeling process and applies it to new data. Usually the optimal box in the peeling process is not the best on unobserved data.

Value

An S3 object of type prim.validate

Author(s)

Jurian Baas

`prim.validate.metrics` *Calculate statistical metrics*

Description

This function calculates the mean, standard deviation, standard error of the mean, 95

Usage

```
prim.validate.metrics(prim.validate)
```

Arguments

`prim.validate`	An object of type "prim validate"

Value

A list with elements described above

Author(s)

Jurian Baas

`quasi.convex.hull` *Calculate a frontier of dominating points*

Description

During the diversify process, we are really only interested in the attempts which dominate all others in performance.

Usage

```
quasi.convex.hull(p.div)
```

Arguments

`p.div`	An object of type "prim.diversify"

Value

A vector of indexes for the dominating points

Author(s)

William Huber & Jurian Baas

See Also

https://stats.stackexchange.com/a/65157

summary.prim.cover *Summarize a PRIM cover result object*

Description

Summarize a PRIM cover result object

Usage

```
## S3 method for class 'prim.cover'
summary(object, ..., round = TRUE, digits = 2)
```

Arguments

object	An S3 object of class prim.cover
...	Optional arguments to pass on
round	Optional setting to disable rounding
digits	Optional setting to control number of digits to round

Value

Nothing, this function is called for its side-effects

Author(s)

Jurian Baas

summary.prim.diversify *Summarize a PRIM diversify object*

Description

Summarize a PRIM diversify result object

Usage

```
## S3 method for class 'prim.diversify'
summary(object, ..., round = TRUE, digits = 2)
```

Arguments

`object`	An S3 object of class prim.diversify
`...`	Optional arguments to pass on
`round`	Optional setting to disable rounding
`digits`	Optional setting to control number of digits to round

Value

Nothing, this function is called for its side-effects

Author(s)

Jurian Baas

`summary.prim.peel` *Summarize a PRIM peeling result object*

Description

Summarize a PRIM peeling result object

Usage

```
## S3 method for class 'prim.peel'
summary(object, ..., round = TRUE, digits = 2)
```

Arguments

`object`	An S3 object of class prim.peel
`...`	Optional arguments to pass on
`round`	Optional setting to disable rounding
`digits`	Optional setting to control number of digits to round

Value

Nothing, this function is called for its side-effects

Author(s)

Jurian Baas

summary.prim.validate *Summarize a PRIM test result object*

Description

Summarize a PRIM test result object

Usage

```
## S3 method for class 'prim.validate'
summary(object, ..., round = TRUE, digits = 2)
```

Arguments

object	An S3 object of class prim.validate
...	Optional arguments to pass on
round	Optional setting to disable rounding
digits	Optional setting to control number of digits to round

Value

Nothing, this function is called for its side-effects

Author(s)

Jurian Baas

Index