

Package ‘rWMBAT’

February 15, 2012

Type Package

Title The William and Mary Bayesian Analysis Tool

Version 2.0

Date 2009-10-28

Author Karl Kuschner, Qian Si and William Cooke, College of William and Mary, Dept. of Physics, 2009

Maintainer Karl Kuschner<kwkusc@wm.edu>

Description The package contains all the functions used to get a resulting Bayesian network.

License GPL (>= 2)

LazyLoad yes

Repository CRAN

Date/Publication 2009-10-29 19:24:14

R topics documented:

rWMBAT-package	2
adjmat	3
adjmat1	3
adjmat2	4
alignedPeakList	4
automi	4
boundary	5
BuildBayesNet	6
ChooseMetaVars	7
clearirrarcs	8
clipclassconnections	8
CondMutualInfo	9
DoTheMath	10
drop	11

ff	11
finaldata	12
findmutualinfos	12
FindProbTables	13
getarcs	14
In	15
InitialProcessing	15
looklr	16
miCalc	17
mi_vc	18
mi_vc_v	18
mi_vv	18
MutualInfo	19
nfold	19
opt2bin	20
opt3bin	21
Options	22
prior	22
ptable	22
repeats	23
Repmat	23
TestCases	24
testgrpbin	25
threshold	25
toListMetaData	25
traingrp	26
traingrpbin	26
traingrpclass	26
vthreshold	26
vvthreshold	27
WMBAT	27
Index	29

Description

The package contains all the functions used to get a resulting Bayesian network. Then main function is "WMBAT", which takes an array of mass spec peak intensities and other information to build and assess a Bayesian network, the primary output is an adjacency matrix describing the resulting Bayesian network. Other functions, e.g., DoTheMath, are sub-functions used in the "WMBAT" to get the final result.

Details

Package: rWMBAT
Type: Package
Version: 1.0
Date: 2009-09-04
License: GPL (>= 2)
LazyLoad: yes

Author(s)

Karl Kuschner, Qian Si and William Cooke, College of William and Mary, Dept. of Physics, 2009

References

<http://kwkusc.people.wm.edu/dissertation/dissertation.htm>

adjmat

Example Data

Description

Logical adjacency matrix

Usage

data(adjmat)

adjmat1

Example Data

Description

Logical adjacency matrix

Usage

data(adjmat1)

`adjmat2`*Example Data*

Description

Logical adjacency matrix

Usage

```
data(adjmat2)
```

`alignedPeakList`*Example Data*

Description

One of the input example data for main function "WMBAT", a list containing information of time labels for peaks and peak intensity values for every spectrum.

Usage

```
data(alignedPeakList)
```

Examples

```
data(alignedPeakList)
names(alignedPeakList)
```

```
plot(alignedPeakList$peaks,alignedPeakList$data[[1]]$Intensities,"l")
#"alignedPeakList$peaks" is an array containing time labels for peaks.
#"alignedPeakList$data[[1]]$Intensities" is an array containing peak intensity values of the 1st spectrum.
```

`automi`*Finds A Threshold For Randomized MI(V C)*

Description

Finds the threshold of a data set's mutual information with a class vector, above which a variable's MI(class, variable) can be expected to be significant.

Usage

```
automi(data, class, repeats)
```

Arguments

data	double array of discrete integer (1:n) values, cases in rows and variables in columns.
class	double column vector, also 1:n. Classification of each case.
repeats	integer, the number of times to repeat the randomization

Details

The threshold for mi (significance level) is found by taking the data set and randomizing the class vector, then calculating MI(CV) for all the variables. This is repeated a number of times. The resulting list of length (repeats *variables) is sorted, and the 99th percentile max MI is taken as the threshold.

Value

a threshold for randomized MI(V C)

Author(s)

Karl Kuschner, Qian Si and William Cooke, College of William and Mary, Dept. of Physics, 2009.

References

<http://kwkusc.people.wm.edu/dissertation/dissertation.htm>

Examples

```
data(traingrpbin, traingrpclass) # load the example input data from package
threshold <- automi( traingrpbin, traingrpclass, 10 )
```

boundary

Example Data

Description

The location used to bin the data to get max MI

Usage

```
data(boundary)
```

 BuildBayesNet

Selects Features And Metafeatures Based On Mutual Info

Description

BuildBayesNet selects features and metafeatures based on mutual info

Usage

```
BuildBayesNet(data, class, ffactor, drop)
```

Arguments

data	integer array containing the data used to build the Bayes net, cases in rows, variables in cols
class	double column vector, the known class variable for each case
ffactor	multiple of auto MI to use to threshold C->V connections
drop	double, MI loss percentage threshold for testing independence. Set to .75 and adjust to filter too few/too many variable-to-variable connections

Details

This function takes a set of training data and an additional variable called "class" and tries to learn a Bayesian Network Structure by examining Mutual Information.

Value

a matrix of zeros and ones, where one in row *i*, column *j* denotes a directed link in a Bayesian network between variable *i* and variable *j*. The class variable is the last row/column.

Note

CALLED FUNCTIONS: automi(finds an MI threshold based on data) findmutualinfos(finds all values MI(VC), MI(VV) and MI(VCIV))

Author(s)

Karl Kuschner, Qian Si and William Cooke , College of William and Mary, Dept. of Physics, 2009.

References

<http://kwkusc.people.wm.edu/dissertation/dissertation.htm>

Examples

```
data(traingrpbin, traingrpclass, ff, drop) #load example data in package
adjmat <- BuildBayesNet( traingrpbin, traingrpclass, ff, drop )
```

ChooseMetaVars

ChooseMetaVars Attempts To Combine Variables into Better Variables

Description

Finds the V-V pairs in the adjacency matrix, and attempts to combine them into a metavariable with a higher mutual information than either variable alone. If it is possible to do this, it returns a new data matrix with the variables combined.

Usage

```
ChooseMetaVars(data, class, adj)
```

Arguments

data	double array of discrete integer values, cases in rows and variables in columns.
class	double column vector. Classification of each case
adj	logical adjacency matrix, variables+1 by variables. Last row is class node. Logical meaning "there is an arc from i to j."

Value

metamatrix	logical whose (i,j) means "variable j was combined into variable i (and erased)"
finaldata	double data matrix with the variable combined and rebinned
leftbound	vector, the new left boundary (vector) for binning
rightbound	vector, the new right boundary (vector) for binning

Note

CALLED FUNCTIONS: opt3bin(rebins combined variables to determine highest MI)

Author(s)

Karl Kushner, Qian Si and William Cooke, College of William and Mary, Dept. of Physics, 2009

References

<http://kwkusc.people.wm.edu/dissertation/dissertation.htm>

Examples

```
data(traingrp, traingrpclass, adjmat) # load example input data in the package
result <- ChooseMetaVars (traingrp, traingrpclass, adjmat)
```

clearirrarcs	<i>Clears Arcs That Are Not C->V Or C->V<->V</i>
--------------	--

Description

Given an adjacency matrix with $V \leftrightarrow V$ arcs in a square matrix and an additional row representing $C \rightarrow V$ (class to variable), this function clears out all $V1 \rightarrow V2$ arcs where $V1$ is not a member of the set of V 's that are class-connected, i.e. have arcs in the final row.

Usage

```
clearirrarcs(adjin)
```

Arguments

adjin	logical array where a true value at position (i,j) means that there is an arc in a directed acyclic graph between (variable) i and variable j.
-------	--

Value

copy of adjin with unneeded arcs cleared

Author(s)

Karl Kushner, Qian Si and William Cooke, College of William and Mary, Dept. of Physics, 2009

References

<http://kwkusc.people.wm.edu/dissertation/dissertation.htm>

Examples

```
data(adjmat1) # load input example data from package
adjout <- clearirrarcs( adjmat1)
```

clipclassconnections	<i>Delinks Variables From Class</i>
----------------------	-------------------------------------

Description

Where two variables are connected to each other and also to the class, attempt to select one as the child of the other and disconnect it from the class. Use $MI(V_i;C|V_j) \ll MI(V_i;C)$ as a test.

Usage

```
clipclassconnections(adj, mivc_vec, mivcv, droptreshold)
```

Arguments

adj	logical matrix where "true" entries at (i,j) mean "an arc exists from the Bayesian network node V_i to V_j ." The class variable C is added at row (number of V 's + 1). "0" values mean no arc.
mi_vc_vec	double row vector containing $MI(C;V_i)$ for each variable
mi_vcv	double array whose (i,j) entry is $MI(V_i,C V_j)$
dropthreshold	double percentage drop from $MI(V_j;C)$ to $MI(V_j;C V_i)$ before declaring that V_i is between C and V_j

Value

copy of adj with the appropriate arcs removed

Author(s)

Karl Kuschner, Qian Si and William Cooke, College of William and Mary, Dept. of Physics, 2009

References

<http://kwkusc.people.wm.edu/dissertation/dissertation.htm>

Examples

```
data(adjmat2,mi_vc,mi_vc_v,drop) #load input example data from package
adjout <- clipclassconnections(adjmat2,t(mi_vc),mi_vc_v,drop)
```

CondMutualInfo	<i>Calculates The Mutual Information Of Two Variables Conditioned On A Third</i>
----------------	--

Description

Calculates the mutual information of two variables conditioned on a third

Usage

```
CondMutualInfo(V1, V2, condV)
```

Arguments

V1	one of the two vectors
V2	one of the two vectors
condV	given condition

Value

mutual information of two variables conditioned on a third

Author(s)

Qian Si, College of William and Mary, Dept. of Physics, 2009

Examples

```
a<-c(1 ,2, 1, 2, 1)
b<-c(2, 1, 2, 1, 1)
condV<-c(2, 1, 2, 2, 1)
MIxyz <- CondMutualInfo(a, b, condV)
```

DoTheMath

Takes A Data Set And Performs Feature Selection

Description

DoTheMath takes a data array, class vector, and other information and builds and assesses a Bayesian network after selecting features from within the data array.

Usage

```
DoTheMath(InputStructure)
```

Arguments

InputStructure a list containing following inputs: **Class:Vector** of length "cases", with discrete values identifying class of each case (may be integer). **ID:double** patient ID array of length cases, with one or more cols. **MZ:vector** of length "variables" holding labels for variables. **Options:logical** 6x1 array. Options are: 1. Normalize on population total ion count (sum across rows). 2. Remove negative data values by setting them to zero. 3. After normalizing, before binning, average cases with same ID. 4. Find the MI threshold by randomization. 5. Take log(data) prior to binning. Negative values set to 1. 6. Remove Low Signal cases. NOT DONE 3 Bin (2 Bin if False). **n:integer**,the "n" in n-fold cross validation. **repeats:integer**,times to repeat the whole process (e.g. re-crossvalidate). **threshold:double**, factor by which the maximum "random" MI us multiplied to find the minimum "significant" MI (double, 1.0-5.0).

Details

This is the umbrella script that loops a specified number of times (see "repeats" above), each time doing a full n-fold cross validation and recording the results. All input and output data are stored in a single data structure, described below.

Value

all the fields of InputStructure, plus, ErrorRate:Vector containing misclassification rate for each repeat.KeyFeatures:Index to vector MZ that identifies features selected

Note

CALLED FUNCTIONS InitialProcessing(Applies the options listed above) BuildBayesNet(Learns a Bayesian Network from the training data) ChooseMetaVars(Combines variables that may not be physically separate molecules) TestCases(Given the BayesNet, tests the "test group" to determine the probability of being in each class) opt3bin(Discretizes continuous data into 3 bins, optimizing MI) FindProbTables(Learns the values P(C,V) for each variable)

Author(s)

Karl Kuschner, Qian Si and William Cooke, College of William and Mary, Dept. of Physics, 2009

References

<http://kwkusc.people.wm.edu/dissertation/dissertation.htm>

Examples

```
data(In) #load input example data from the package
OutputDataStructure <- DoTheMath (In) ## Running this example may take some time, about 10 minutes on a Intel(R) Core
```

drop

Example Data

Description

double vector, MI loss percentage threshold

Usage

```
data(drop)
```

ff

Example Data

Description

multiple of auto MI to use to threshold C->V connections

Usage

```
data(ff)
```

finaldata

Example Data

Description

double array of discrete integer values, cases in rows and variables in columns

Usage

```
data(finaldata)
```

findmutualinfos

Finds Various Mutual Info Combos among Variables

Description

Given a set of data (many cases, each with values for many variables) and an additional value stored in the vector class, it finds MI described below in "Value."

Usage

```
findmutualinfos(data, class)
```

Arguments

data	a number of cases (in rows), each with a measurement for a group of variables (in columns). The data should be discredited into integers 1 through k. The columns are considered variables V1, V2, ...
class	a column vector of length "cases" with integer values 1,2..., an additional measurement of class C

Value

mi_vc	a row double vector whose ith value is $MI(V_i, C)$
mi_vv	double symmetric matrix with values $MI(V_i, V_j)$
mi_vc_v	double nonsym matrix with values $MI(V_i, C; V_j)$

Note

CALLED FUNCTIONS MutualInfo(Calculate Mutual Information Of Two Variables) CondMutualInfo(Calculate Mutual Information of Two Variables Conditioned On a Third)

Author(s)

Karl Kuschner, Qian Si and William Cooke, College of William and Mary, Dept. of Physics, 2009

References

<http://kwkusc.people.wm.edu/dissertation/dissertation.htm>

Examples

```
data(traingrpbin, traingrpclass) # load example input data from package
result <- findmutualinfos(traingrpbin, traingrpclass)
```

FindProbTables	<i>Estimates The Probabilities $P(\text{class}=c \text{data}=D)$</i>
----------------	---

Description

Input a training group of data arranged with cases in rows and variables in columns, as well as the class value c for that vector. Each case represents a data vector V . For each possible data value v_i , and each variable V_i , it calculates $P(C=c|V_i=v_i)$ and stores that result in a 3-D table. The table is arranged with the dimensions (class value, data value, variable number).

Usage

```
FindProbTables(data, class)
```

Arguments

data	double array of discrete integer values, cases in rows and variables in columns
class	double column vector. Classification of each case

Value

3-D array whose (c,d,v) value is $P(\text{class}=c|\text{data}=p)$ for variable v

Author(s)

Karl Kushner, Qian Si and William Cooke, College of William and Mary, Dept. of Physics, 2009

References

<http://kwkusc.people.wm.edu/dissertation/dissertation.htm>

Examples

```
data(finaldata, traingrpclass) #load the example input data from package
prohtable <- FindProbTables(finaldata, traingrpclass)
```

 getarcs

Builds The Adjacency Matrix For A Set Of Variables

Description

By comparing mutual information between two variables to thresholds determined separately, this function declares there to be an arc in a Bayesian network. Arcs are stored in an adjacency matrix, described below.

Usage

```
getarcs(mvc, vcthreshold, mvv, vvthreshold)
```

Arguments

mvc	double vector array with mutual information between variables and the class (variables and other variables). The (i,j) entries of mvv are $MI(V_i, V_j)$
vcthreshold	scalar threshold used to test for existence linkz
mvv	double vector matrix
vvthreshold	scalar threshold used to test for existence linkz

Details

The primary tests are $MI[V_i, C] \gg vcthreshold$ (tests for links between V_i and the class) $MI[V_i, V_j] \gg vvthreshold$ (tests the links between variables)

Value

logical matrix whose entries "1" at (i,j) mean "an arc exists from the Bayesian network node V_i to V_j ." The class variable C is added at row (number of V's + 1). "0" values mean no arc

Author(s)

Karl Kushner, Qian Si and William Cooke, College of William and Mary, Dept. of Physics, 2009

References

<http://kwkusc.people.wm.edu/dissertation/dissertation.htm>

Examples

```
data(mi_vc, vcthreshold, mi_vv, vvthreshold) # load the example input data from the package
adjmat<-getarcs(mi_vc, vcthreshold, mi_vv, vvthreshold)
```

In	<i>Example Data</i>
----	---------------------

Description

The input example data for function DoTheMath. A list containing information about MZ,ID,Intensities,Class of spectra and Options,

Usage

data(In)

InitialProcessing	<i>INITIALPROCESSING Initial Prep of Data from Signal Processing</i>
-------------------	--

Description

Takes peak list which has been imported into R and prepares the peak list For Bayesian Analysis

Usage

InitialProcessing(StructIn)

Arguments

StructIn	List with the following double-typed arrays: Intensities:double n x m real-valued array with variables (peaks) in columns, cases (samples) in rows. MZ:double list of the labels (m/z value) for each of the variables.Must be the same size as the number of variables in Intensities. Class:intefer vector, classification of each sample (disease state)– 1 or 2–must be the same size as the number of cases in Intensities. ID:double column array, case or patient ID number, same size as class. May have second column, so each row is [ID1 ID2] where ID2 is replicate number. Options:logical Array of processing options with elements 1. Normalize. 2. Clip Data (remove negatives). 3. Replicate Average. 4. Auto threshold MI. 5. Use Log of Data. 6. Remove Low Signal cases. NOT DONE 3 Bin (2 Bin if False).
----------	---

Value

RawData	Intensities as input
ClipData	RawData where all values less than 1 are set to 1
NormData	ClipData normalized by total ion count, i.e. divided by the sum of all variables for each case
LogData	Natural logarithm of NormData

Class	Same as input
MZ	Same as input
ID	Single column. If replicates are not averaged, the entries are now ID1.ID2. If replicates averaged, then just ID1
DeltaMZ	difference in peak m/z values to look for adducts
RatioMZ	ratios of m/z values ot look for satellites

Author(s)

Karl Kuschner, Qian Si and William Cooke, College of William and Mary, Dept. of Physics, 2009

References

<http://kwkusc.people.wm.edu/dissertation/dissertation.htm>

Examples

```
data(In) #load input example data from the package
StructOut <- InitialProcessing(In)
```

looklr	<i>Finds Boundary</i>
--------	-----------------------

Description

Given a start position, finds another boundary (to create 3 bins) that maximizes MI with the class.

Usage

```
looklr(data, class, startbd, steps)
```

Arguments

data	double array, cases in rows and variables in columns
class	double column vector, values 1:c representing classification of each case
startbd	double vector, given start position for each case
steps	integer, number of steps to test at while finding maximum MI

Value

miout	double vector, recorded highest MI value
nextboundary	double vector, boundary (to create 3 bins) that maximizes MI with the class
binned	double matrix, recorded the binned value

Author(s)

Karl Kushner, Qian Si and William Cooke, College of William and Mary, Dept. of Physics, 2009

References

<http://kwkusc.people.wm.edu/dissertation/dissertation.htm>

Examples

```
data(traingrp,traingrpclass,boundary)#load the example data input data from package
result<- looklr (traingrp,traingrpclass,boundary, 150)
```

miCalc

C code to Speed the Calculation of Mutual Information

Description

C code to Speed the Calculation of Mutual Information

Usage

```
miCalc(v1, v2, n, MI)
```

Arguments

v1	one of the two vectors
v2	one of the two vectors
n	length of the vector
MI	mutual information of the two vectors

Value

mutual information of the two vectors

Author(s)

Bill Cooke, College of William and Mary, Dept. of Physics, 2009

`mi_vc`*Example Data*

Description

double vector array with mutual information between variables and the class (variables and other variables). The (i,j) entries of mvv are $MI(V_i, V_j)$.

Usage`data(mi_vc)`

`mi_vc_v`*Example Data*

Description

example data

Usage`data(mi_vc_v)`

`mi_vv`*Example Data*

Description

double vector matrix

Usage`data(mi_vv)`

`MutualInfo`*Calculates Mutual Information Of The Two Variables*

Description

Calculates mutual information of the two variables

Usage

```
MutualInfo(v1, v2)
```

Arguments

<code>v1</code>	one of the two vectors
<code>v2</code>	one of the two vectors

Value

mutual information of the two vectors

Author(s)

Bill Cooke, College of William and Mary, Dept. of Physics, 2009

Examples

```
a<-c(1 ,2, 1, 2, 1)
b<-c(2, 1, 2, 1, 1)
MI <- MutualInfo(a, b)
```

`nfold`*Example Data*

Description

the "n" in n-fold cross validation (integer 4-10). 4 is used here as input example data, but 10 is recommended when processing real data.

Usage

```
data(nfold)
```

Examples

```
data(nfold)
```

 opt2bin

Finds The Best Single Boundary For Each Variable To Maximize MI

Description

This function takes an array of continuous data, with cases in rows and variables in columns, along with a vector "class" which holds the known class of each of the cases, and returns an array "binned-data" that holds the 2 bin discretized data.

Usage

```
opt2bin(rawdata, class, steps, typesearch, minint = NA, maxint = NA)
```

Arguments

rawdata	double array of continuous values, cases in rows and variables in columns. Distribution is unknown
class	double column vector, values 1:c representing classification of each case
steps	integer, number of steps to test at while finding maximum MI
typesearch	=0 starting bndry based on data's actual max/min values =1 use the value passed in max as maximum (right) value =-1 use the value passed in min as minimum (left) value =2 used values passed via max, min
minint	vectors whose values limit the range of search for each variables boundaries
maxint	vectors whose values limit the range of search for each variables boundaries

Details

The discretization bin boundary is found by maximizing the mutual information with the class the resulting MI and boundary are also returned. The starting boundaries for the search can be given in the vectors min and max, or either one, or neither, in which case the data values determine the search boundaries.

Value

mi	row vector holding the maximum values of MI(CVi) found
boundary	double vector, the location used to bin the data to get max MI
binneddata	resulting data binned into "1" (low) or "2" (hi)

Author(s)

Karl Kushner, Qian Si and William Cooke, College of William and Mary, Dept. of Physics, 2009

References

<http://kwkusc.people.wm.edu/dissertation/dissertation.htm>

Examples

```
data(traingrp,traingrpclass) #load example input data from package
result <- opt2bin(traingrp,traingrpclass,150,2)
```

 opt3bin

Find 3 Bin Boundaries Optimizing the MI of Each Variable

Description

This function takes an array of continuous sample data of size cases (rows) by variables (columns), along with a class vector of integers, each integer specifying the class. The class vector has the same number of cases as the data. The function outputs the position of the 2 bin boundaries (3 bins) that optimize the mutual information of each variable's data vector with the class vector.

Usage

```
opt3bin(data, class)
```

Arguments

data	double array of continuous values, cases in rows and variables in columns. Distribution is unknown
class	double column vector, values 1:c representing classification of each case

Value

l	double row vector of left boundary position for each var
r	double row vector of right boundary position for each var
binned	double data array discretized using boundaries in l and r
mi	double row vector of mutual info between each discr. variable and class

Author(s)

Karl Kuschner, Qian Si and William Cooke, College of William and Mary, Dept. of Physics, 2009

References

<http://kwkusc.people.wm.edu/dissertation/dissertation.htm>

Examples

```
data(traingrp,traingrpclass) #load example data in package
result <- opt3bin(traingrp,traingrpclass)
```

Options

Example Data

Description

Logical 6x1 array

Usage

data(Options)

Details

Options are: 1. Normalize on population total ion count (sum across rows) 2. Remove negative data values by setting them to zero 3. After normalizing, before binning, average cases with same ID 4. NOT USED - SET TO FALSE 5. Take log(data) prior to binning. Negative values set to 1. 6. NOT USED - SET TO FALSE

Examples

data(Options)

prior

Example Data

Description

double columns vector, prior probabilities for each cases in data

Usage

data(prior)

ptable

Example Data

Description

3-D double array of probabilities

Usage

data(ptable)

repeats	<i>Example Data</i>
---------	---------------------

Description

Integer, times to repeat the whole process (e.g.re-crossvalidate). 2 is used here for the example data, but 100 is recommended when processing the real data

Usage

```
data(repeats)
```

Examples

```
data(repeats)
```

Repmat	<i>Creates A Large Matrix Consisting Of An m-by-n Tiling Of Copies Of x</i>
--------	---

Description

Creates a large matrix consisting of an m-by-n tiling of copies of X

Usage

```
Repmat(X, m, n)
```

Arguments

X	matrix used to make copies of
m	row copy number of the result matrix
n	column copy number of the result matrix

Value

matrix, m-by-n tiling of copies of input matrix

Author(s)

Qian Si, College of William and Mary, Dept. of Physics, 2009

Examples

```
Repmat(diag(3),2,3)  
# diag(3) creates a 3-by-3 identity matrix
```

 TestCases

Uses Bayes Rule To Classify A Case

Description

Tests each of a set of data vectors by looking up $P(\text{data}|\text{class})$ in a probability table, then finding $P(\text{case}|\text{class})$ by multiplying each of those values in a product. Then uses Bayes' rule to calculate $P(\text{class}|\text{data})$ for each possible value of class. Reports this as an array of class probabilities for each case.

Usage

```
TestCases(p, prior, data)
```

Arguments

p	3-D double array of probabilities (c,d,v). The first dimension is the class, the second is the data value, the third is the variable number. The entry is $P(\text{var } v=\text{value } d \mid \text{class}=\text{value } c)$
prior	double columns vector, prior probabilities for each cases in data
data	double array of discrete integer (1:n) values, cases in rows and variables in columns

Value

2-D double array whose value is $P(\text{class}=\text{c}|\text{data})$ for each case. Cases are in rows, class in cols

Author(s)

Karl Kuschner, Qian Si and William Cooke, College of William and Mary, Dept. of Physics, 2009

References

<http://kwkusc.people.wm.edu/dissertation/dissertation.htm>

Examples

```
data(ptable, prior, testgrpbin) # load example input data from package
classprohtable <- TestCases (ptable, prior, testgrpbin)
```

testgrpbin	<i>Example Data</i>
------------	---------------------

Description

double array of discrete integer values, cases in rows and variables in columns

Usage

```
data(testgrpbin)
```

threshold	<i>Example Data</i>
-----------	---------------------

Description

Factor by which the maximum "random" MI is multiplied to find the minimum "significant" MI (double, 1.0-5.0)

Usage

```
data(threshold)
```

Details

recommend starting with 1 and increasing until a "reasonable" number of diagnostic peaks is reached and error rates are minimized. This setting is dependant on the data and the correlations between variables.

tofListMetaData	<i>Example Data</i>
-----------------	---------------------

Description

One of the input example data for main function "WMBAT", a list containing information about every spectra, the information related to the spectra we needed in this package is class and ID. Class,Integer vector, values 1 or 2 identifying the class of each case, such as "disease, non-disease" ID,Double one or two column array contains the sample IDfor each case. Second column is optional and would identifyreplicates of the same sample.

Usage

```
data(tofListMetaData)
```

traingrp

Example Data

Description

double array of discrete integer (1:n) values, cases in rows and variables in columns

Usage

data(traingrp)

traingrpbm

Example Data

Description

integer array containing the data used to build the Bayes net

traingrpclass

Example Data

Description

double column vector. Classification of each case

Usage

data(traingrpclass)

vcthreshold

Example Data

Description

scalar threshold used to test for existence linkz

vvthreshold

Example Data

Description

scalar threshold used to test for existence linkz

Usage

data(vvthreshold)

WMBAT

The William and Mary Bayesian Analysis Tool

Description

WMBAT takes an array of mass spec peak intensities, a vector describing which of two classes each sample belongs to, and other information and builds and assesses a Bayesian network after selecting features (peaks) from within the data array that are diagnostic of the class. The primary output is an adjacency matrix describing the resulting Bayesian network.

Usage

WMBAT(tofListMetaData, alignedPeakList, Options, nfold, repeats, threshold)

Arguments

- alignedPeakList
contain information about mass spec peak intensity values
- tofListMetaData
a list containing information about every spectra, the information related to the spectra we needed in this package is class and ID. Class,Integer vector, values 1 or 2 identifying the class of each case, such as "disease, non-disease" ID,Double one or two column array contains the sample IDfor each case. Second column is optional and would identifyreplicates of the same sample.
- Options
Logical 6x1 array. Options are 1. Normalize on population total ion count (sum across rows) 2. Remove negative data values by setting them to zero 3. After normalizing, before binning, average cases with same ID 4. NOT USED - SET TO FALSE 5. Take log(data) prior to binning. Negative values set to 1. 6. NOT USED - SET TO FALSE
- nfold
integer, the "n" in n-fold cross validation (integer 4-10). 10 is recommended
- repeats
integer, times to repeat the whole process (e.g.re-crossvalidate). 100 is recommended

threshold double, factor by which the maximum "random" MI is multiplied to find the minimum "significant" MI (double, 1.0-5.0). We recommend starting with 1 and increasing until a "reasonable" number of diagnostic peaks is reached and error rates are minimized. This setting is dependant on the data and the correlations between variables

Value

IntOut double Intensities input array, after processing by the various options selected by the logical Options above

IDOut double vector, the ID number of each row in the IntOut array. With no replicate averaging, each ID will be preserved (but reformatted) from the input. With replicate averaging, only the primary ID number remains.

PredClass double matrix, the predicted class of each case, during each of the trials (from input "repeats")

Class2Vars vector whose ith value is the fraction of times peak i (from the vector MZ) was selected as being connected to the class. The maximum times it could have been selected was nfold*repeats, use which (Class2Vars > 0.5) to find those variables selected more than 0.5 = 50 percent

Var2Vars integer array whose (i,j) entry is the fraction of times a second level link was found from peak i to peak j, when peak i was connected to the class, as found in SumLv11

MetaVars integer array whose (i,j) entry is the fraction of times a metavariable was created using peak i and peak j and stored in the level

TrialErr double vector, the error rate for each of the "repeats" possible trials. Records the percentage of cases where PredClass was not equal to the input Class.

Note

CALLED FUNCTIONS DoTheMath(Learns a Bayesian Network from the data)

Author(s)

Karl Kuschner, Qian Si and William Cooke, College of William and Mary, Dept. of Physics, 2009

References

<http://kwkusc.people.wm.edu/dissertation/dissertation.htm>

Examples

```
data(tofListMetaData, alignedPeakList, Options, nfold, repeats, threshold)#load input example data from the package
result <- WMBAT (tofListMetaData, alignedPeakList, Options, nfold, repeats, threshold)# Running this example may take
```

Index

*Topic **datasets**

- adjmat, 3
 - adjmat1, 3
 - adjmat2, 4
 - alignedPeakList, 4
 - boundary, 5
 - drop, 11
 - ff, 11
 - finaldata, 12
 - In, 15
 - mi_vc, 18
 - mi_vc_v, 18
 - mi_vv, 18
 - nfold, 19
 - Options, 22
 - prior, 22
 - ptable, 22
 - repeats, 23
 - testgrpbin, 25
 - threshold, 25
 - toListMetaData, 25
 - traingrp, 26
 - traingrpbin, 26
 - traingrpclass, 26
 - vcthreshold, 26
 - vvthreshold, 27
-
- adjmat, 3
 - adjmat1, 3
 - adjmat2, 4
 - alignedPeakList, 4
 - automi, 4
-
- boundary, 5
 - BuildBayesNet, 6
-
- ChooseMetaVars, 7
 - clearirrarcs, 8
 - clipclassconnections, 8
 - CondMutualInfo, 9
-
- DoTheMath, 10
 - drop, 11
-
- ff, 11
 - finaldata, 12
 - findmutualinfos, 12
 - FindProbTables, 13
-
- getarcs, 14
-
- In, 15
 - InitialProcessing, 15
-
- looklr, 16
-
- mi_vc, 18
 - mi_vc_v, 18
 - mi_vv, 18
 - miCalc, 17
 - MutualInfo, 19
-
- nfold, 19
-
- opt2bin, 20
 - opt3bin, 21
 - Options, 22
-
- prior, 22
 - ptable, 22
-
- repeats, 23
 - Repmat, 23
 - rWMBAT (rWMBAT-package), 2
 - rWMBAT-package, 2
-
- TestCases, 24
 - testgrpbin, 25
 - threshold, 25
 - toListMetaData, 25
 - traingrp, 26
 - traingrpbin, 26

traingrpclass, [26](#)

vcthreshold, [26](#)

vvthreshold, [27](#)

WMBAT, [27](#)