Package ‘symbolicDA’

March 14, 2015

Title Analysis of Symbolic Data
Version 0.4-2
Date 2015-03-13
Author Andrzej Dudek, Marcin Pelka <marcin.pelka@ue.wroc.pl>, Justyna Wilk<justyna.wilk@ue.wroc.pl>
Maintainer Andrzej Dudek <andrzej.dudek@ue.wroc.pl>
Depends clusterSim,XML
Imports rgl,shapes,e1071,ade4,cluster
Description Symbolic data analysis methods: importing/exporting data from ASSO XML Files, distance calculation for symbolic data (Ichino-Yaguchi, de Carvalho measure), zoom star plot, 3d interval plot, multidimensional scaling for symbolic interval data, dynamic clustering based on distance matrix, HINoV method for symbolic data, Ichino's feature selection method, principal component analysis for symbolic interval data, decision trees for symbolic data based on optimal split with bagging, boosting and random forest approach (+visualization), kernel discriminant analysis for symbolic data, Kohonen's self-organizing maps for symbolic, replication and profiling, artificial symbolic data generation.
License GPL (>= 2)
URL http://keii.ue.wroc.pl/symbolicDA
Encoding UTF-8
NeedsCompilation yes
Repository CRAN
Date/Publication 2015-03-14 08:17:18

R topics documented:

  bagging.SDA ............................................... 2
  boosting.SDA ............................................... 4
  cars .......................................................... 5
  cluster.Description.SDA ..................................... 6
  data_symbolic ............................................... 8
  DClust ......................................................... 8
Bagging algorithm for optimal split based on decision tree for symbolic objects

### Usage

```r
bagging.SDA(sdt, formula, testSet, mfinal = 20, rf = FALSE, ...)
```
Arguments

- **sdt**: Symbolic data table
- **formula**: formula as in `ln` function
- **testSet**: a vector of integers indicating classes to which each objects are allocated in learning set
- **mfinal**: number of partial models generated
- **rf**: random forest like drawing of variables in partial models
- **...**: arguments passed to `decisionTree.SDA` function

Details

The bagging, which stands for bootstrap aggregating, was introduced by Breiman in 1996. The diversity of classifiers in bagging is obtained by using bootstrapped replicas of the training data. Different training data subsets are randomly drawn with replacement from the entire training data set. Then each training data subset is used to train a decision tree (classifier). Individual classifiers are then combined by taking a simple majority vote of their decisions. For any given instance, the class chosen by most number of classifiers is the ensemble decision.

Value

An object of class `bagging.SDA`, which is a list with the following components:

- **predclass**: the class predicted by the ensemble classifier
- **confusion**: the confusion matrix for ensemble classifier
- **error**: the classification error
- **pred**: ?
- **classfinal**: final class memberships

Author(s)

Andrzej Dudek <andrzej.dudek@ue.wroc.pl> Marcin Pelka <marcin.pelka@ue.wroc.pl>

Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland

http://keii.ue.wroc.pl/symbolicDA

References


See Also

boosting.SDA, random.forest.SDA, decisionTree.SDA

Examples

Example will be available in next version of package, thank You for your patience :-)
Value

- **formula**: a symbolic description of the model that was used
- **trees**: trees built while making the ensemble
- **weights**: weights for each object from test set
- **votes**: final consensus clustering
- **class**: predicted class memberships
- **error**: error rate of the ensemble clustering

Author(s)

Andrzej Dudek <andzej.dudek@ue.wroc.pl> Marcin Pelka <marcin.pelka@ue.wroc.pl>
Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland
http://keii.ue.wroc.pl/symbolicDA

References


See Also

bagging.SDA, random.forest.SDA, decisionTree.SDA

Examples

#Example will be available in next version of package, thank you for your patience :-)

<table>
<thead>
<tr>
<th>cars</th>
<th>real data set in symbolic form - selected car models described by a set of symbolic variables</th>
</tr>
</thead>
</table>

Description

symbolic data set: 30 observations on 12 symbolic variables - 9 interval-valued and 3 multinominal variables, third dimension represents the beginning and the end of intervals for interval-valued variable’s implementation or a set of categories for multinominal variable’s implementation

Format

symbolic data table (see (link(symbolic.object)))
Source

the original data on 30 selected car models and their prices, chasis and engine types were collected from the websites of authorized car dealers. Then the data were converted (aggregated) to symbolic format (second order symbolic objects). Each symbolic object - e.g. "Seat Leon", "Citroen C4" - represents all chasis, engine types and price range of this kind of car model available on the Polish market in 2010. For example the price range [54,900; 96,190] PLN, hatchback and saloon body style, petrol and diesel engine, acceleration 0-100 kph range [10.00; 11.90] seconds are, in general, the characteristics of "Toyota Corolla".

Examples

# LONG RUNNING - UNCOMMENT TO RUN
#data("cars",package="symbolicDA")
#sdt<-cars
#r<- HINoV.SDA(sdt, u=5, distance="U_3")
#print(r$stopri)
#plot(r$stopri[,2], xlab="Variable number", ylab="topri",
#xaxt="n", type="b")
#axis(1,at=c(1:max(r$stopri[,1])),labels=r$stopri[,1])

car.SDA

description of clusters of symbolic objects

description of clusters of symbolic objects is obtained by a generalisation operation using in most cases descriptive statistics calculated separately for each cluster and each symbolic variable.

Usage

cluster.SDA(table.Symbolic, clusters, precission=3)

Arguments

table.Symbolic  Symbolic data table
clusters         a vector of integers indicating the cluster to which each object is allocated
precission       Number of digits to round the results

Value

A List of cluster numbers, variable number and labels.
The description of clusters of symbolic objects which differs according to the symbolic variable type:
- for interval-valued variable:
"min value" - minimum value of the lower-bounds of intervals observed for objects belonging to the cluster
"max value" - maximum value of the upper-bounds of intervals observed for objects belonging to the cluster
- for multinominal variable:
"categories" - list of all categories of the variable observed for symbolic belonging to the cluster
- for multinominal with weights variable:
"min probabilities" - minimum weight of each category of the variable observed for objects belonging to the cluster
"max probabilities" - maximum weight of each category of the variable observed for objects belonging to the cluster
"avg probabilities" - average weight of each category of the variable calculated for objects belonging to the cluster
"sum probabilities" - sum of weights of each category of the variable calculated for objects belonging to the cluster

Author(s)
Andrzej Dudek <andrzej.dudek@ue.wroc.pl>, Justyna Wilk <justyna.wilk@ue.wroc.pl> Department of Econometrics and Computer Science, Wroclaw University of Economics, Poland http://kei.ue.wroc.pl/symbolicDA

References

See Also
SClust,DClust; hclust in stats library; pam in cluster library

Examples
# LONG RUNNING - UNCOMMENT TO RUN
#data("cars", package="symbolicDA")
#y<-cars
#cl<-SClust(y, 4, iter=150)
#print(cl)
#o<-cluster.Description.SDA(y, cl)
#print(o)
data_symbolic  Symbolic interval data

**Description**
Artificially generated symbolic interval data

**Format**
3-dimensional array: 125 objects, 6 variables, third dimension represents beginning and end of interval, 5-class structure

**Source**
Artificially generated data

**Examples**
```r
library(symbolicDA)
data(data_symbolic)
r <- HINoV.SDA(simple2SD(data_symbolic), u=5)
print(r$stopri)
plot(r$stopri[,2], xlab="Variable number", ylab="topri",
xaxt="n", type="b")
axis(1, at=c(1:max(r$stopri[,1])), labels=r$stopri[,1])
```

DClust  Dynamical clustering based on distance matrix

**Description**
Dynamical clustering of objects described by symbolic and/or classic (metric, non-metric) variables based on distance matrix

**Usage**
```r
DClust(dist, cl, iter=100)
```

**Arguments**
- `dist` distance matrix
- `cl` number of clusters or vector with initial prototypes of clusters
- `iter` maximum number of iterations

**Details**
See file `../doc/DClust_details.pdf` for further details
Value

A vector of integers indicating the cluster to which each object is allocated.

Author(s)

Andrzej Dudek <andrzej.dudek@ue.wroc.pl>, Justyna Wilk <justyna.wilk@ue.wroc.pl> Department of Econometrics and Computer Science, Wrocław University of Economics, Poland http://keii.ue.wroc.pl/symbolicDA

References


See Also

Sclust, dist.SDA; dist in stats library; dist.GDM in clustersim library; pam in cluster library

Examples

```r
# LONG RUNNING - UNCOMMENT TO RUN
#data("cars", package="symbolicDA")
#sdt<-cars
#dist<-dist.SDA(sdt, type="U_3")
#clust<-DClust(dist, cl=5, iter=100)
#print(clust)
```

decisionTree.SDA Decison tree for symbolic data

Description

Optimal split based decision tree for symbolic objects

Usage

decisionTree.SDA(sdt, formula, testSet, threshMin=0.0001, threshW=-1e10, tNodes=NULL, minSize=2, epsilon=1e-4, useEM=FALSE, multiNominalType="ordinal", rf=FALSE, rf.size, objectSelection)
Arguments

- **sdt**: Symbolic data table
- **formula**: formula as in ln function
- **testSet**: a vector of integers indicating classes to which each objects are allocated in learnig set
- **threshMin**: parameter for tree creation algorithm
- **threshW**: parameter for tree creation algorithm
- **tNodes**: parameter for tree creation algorithm
- **minSize**: parameter for tree creation algorithm
- **epsilon**: parameter for tree creation algorithm
- **useEM**: use Expectation Optimalization algorithm for estimating conditional probabilities
- **multiNominalType**: "ordinal" - functione treats multi-nominal data as ordered or "nominal" functione treats multi-nomianal data as unordered (longer performance times)
- **rf**: if TRUE symbolic variables for tree creation are randomly chosen like in random forest algorithm
- **rf.size**: the number of variables chosen for tree creation if rf is true
- **objectSelection**: optional, vector with symbolic object numbers for tree creation

Details

For further details see ../doc/decisionTree_SDA.pdf

Value

- **nodes**: nodes in tree
- **nodeObjects**: contribution of each objects nodes in tree
- **conditionalProbab**: conditional probability of belongingness of nodes to classes
- **prediction**: predicted classes for objects from testSet

Author(s)

Andrzej Dudek <andrzej.dudek@ue.wroc.pl> Marcin Pelka <marcin.pelka@ue.wroc.pl>

Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland

http://keii.ue.wroc.pl/symbolicDA
dist.SDA

References


See Also

bagging.SDA, boosting.SDA, randomForest.SDA, draw.decisionTree.SDA

Examples

```r
# Example 1
# LONG RUNNING - UNCOMMENT TO RUN
# File samochody.xml needed in this example
# can be found in /inst/xml library of package
table<parse.SO("samochody")
tree<decisionTree.SDA(sda, "Typ_samochodu-.", testSet=1:33)
summary(tree) # a very gerneral information
#tree # summary information
```

dist.SDA  

*distance measurement for symbolic data*

Description

calculates distances between symbolic objects described by interval-valued, multinominal and multinominal with weights variables

Usage

dist.SDA(table.Symbolic,type="U_2",subType=NULL,gamma=0.5,power=2,probType="J", probAggregation="P_1",s=0.5,p=2,variableSelection=NULL,weights=NULL)

Arguments

table.Symbolic  symbolic data table
type  distance measure for boolean symbolic objects: H, U_2, U_3, U_4, C_1, SO_1, SO_2, SO_3, SO_4, SO_5; mixed symbolic objects: L_1, L_2
subType  comparison function for C_1 and SO_1: D_1, D_2, D_3, D_4, D_5
gamma  gamma parameter for U_2 and U_3, gamma [0, 0.5]
power  power parameter for U_2 and U_3; power [1, 2, 3, ..]
probType  distance measure for probabilistic symbolic objects: J, CHI, REN, CHER, LP
probAggregation

aggregation function for J, CHI, REN, CHER, LP: P_1, P_2

s

parameter for Renyi (REN) and Chernoff (CHE) distance, s [0, 1)

p

parameter for Minkowski (LP) metric; p=1 - manhattan distance, p=2 - euclidean distance

variableSelection

numbers of variables used for calculation or NULL for all variables

weights

weights of variables for Minkowski (LP) metrics

Details

Distance measures for boolean symbolic objects:
H - Hausdorff’s distance for objects described by interval-valued variables, U_2, U_3, U_4 - Ichino-Yaguchi’s distance measures for objects described by interval-valued and/or multinominal variables, C_1, SO_1, SO_2, SO_3, SO_4, SO_5 - de Carvalho’s distance measures for objects described by interval-valued and/or multinominal variables.

Distance measurement for probabilistic symbolic objects consists of two steps: 1. Calculation of distance between objects for each variable using componentwise distance measures: J (Kullback-Leibler divergence), CHI (Chi-2 divergence), REN (Renyi’s divergence), CHER (Chernoff’s distance), LP (modified Minkowski metrics). 2. Calculation of aggregative distance between objects based on componentwise distance measures using objectwise distance measure: P_1 (manhattan distance), P_2 (euclidean distance).

Distance measures for mixed symbolic objects - modified Minkowski metrics: L_1 (manhattan distance), L_2 (euclidean distance).

See file ../doc/dist_SDA.pdf for further details

Value

distance matrix of symbolic objects

Author(s)

Andrzej Dudek <andrzej.dudek@ue.wroc.pl>, Justyna Wilk <justyna.wilk@ue.wroc.pl> Department of Econometrics and Computer Science, Wroclaw University of Economics, Poland https://keii.ue.wroc.pl/symbolicDA

References

Diday E., Noirhomme-Fraiture M. (Eds.) (2008), Symbolic Data Analysis with SODAS Software, John Wiley & Sons, Chichester.

**See Also**

*DClust*, *indexGl*, *dist.Symmetric* in *clusterSim* library

**Examples**

```R
# LONG RUNNING - UNCOMMENT TO RUN
#data("cars", package="symbolicDA")
#dist<-dist.SDA(cars, type="U_3", gamma=0.3, power=2)
#print(dist)
```

---

**draw.decisionTree.SDA**  
*Draws optimal split based decision tree for symbolic objects*

**Description**

Draws optimal split based decision tree for symbolic objects

**Usage**

```
draw.decisionTree.SDA(decisionTree.SDA, boxWidth=1, boxHeight=3)
```

**Arguments**

- `decisionTree.SDA`: optimal split based decision tree for symbolic objects (result of `decisionTree.SDA` function)
- `boxWidth`: width of single box in drawing
- `boxHeight`: height of single box in drawing

**Details**

Draws optimal split based decision (classification) tree for symbolic objects.

**Value**

A draw of optimal split based decision (classification) tree for symbolic objects.

**Author(s)**

Andrzej Dudek <andrzej.dudek@ue.wroc.pl>  
Marcin Pelka <marcin.pelka@ue.wroc.pl>  
Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland  
[http://keii.ue.wroc.pl/symbolicDA](http://keii.ue.wroc.pl/symbolicDA)
References


See Also
decisionTree.SDA

Examples

# LONG RUNNING - UNCOMMENT TO RUN
# Files samochody.xml and wave.xml needed in this example
# can be found in /inst/xml library of package

# Example 1
#sda<-parse.SO("samochody")
#tree<decisionTree.SDA(sda, "Typ_samochodu:.", testSet=26:33)
#draw.decisionTree.SDA(tree,boxWidth=1,boxHeight=3)

# Example 2
#sda<-parse.SO("wave")
#tree<decisionTree.SDA(sda, "Waveform:.", testSet=1:30)
#draw.decisionTree.SDA(tree,boxWidth=2,boxHeight=3)

---

generate.SO generation of artificial symbolic data table with given cluster structure

Description

generation of artificial symbolic data table with given cluster structure

Usage

generate.SO(numObjects,numClusters,numIntervalVariables,numMultivaluedVariables)

Arguments

numObjects number of objects in each cluster
numClusters number of objects
numIntervalVariables Number of symbolic interval variables in generated data table
numMultivaluedVariables Number of symbolic multi-valued variables in generated data table
Value

data  symbolic data table with given cluster structure
clusters  vector with cluster numbers for each object

Author(s)

Andrzej Dudek <andrzej.dudek@ue.wroc.pl>

Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland
http://keii.ue.wroc.pl/symbolicDA

References


See Also

see symbolic.object for symbolic data table R structure representation

Examples

# Example will be available in next version of package, thank You for your patience :-)

---

HINoV.SDA  Modification of HINoV method for symbolic data

Description

Carmone, Kara \& Maxwell’s Heuristic Identification of Noisy Variables (HINoV) method for symbolic data

Usage

HINoV.SDA(table.Symbolic, u=NULL, distance="H", Index="cRAND", method="pam", ...)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>table.Symbolic</td>
<td>symbolic data table</td>
</tr>
<tr>
<td>u</td>
<td>number of clusters</td>
</tr>
<tr>
<td>distance</td>
<td>symbolic distance measure as parameter type in <code>dist.SDA</code></td>
</tr>
<tr>
<td>method</td>
<td>clustering method: &quot;single&quot;, &quot;ward&quot;, &quot;complete&quot;, &quot;average&quot;, &quot;mcquitty&quot;, &quot;median&quot;, &quot;centroid&quot;, &quot;pam&quot; (default), &quot;SClust&quot;, &quot;DClust&quot;</td>
</tr>
<tr>
<td>Index</td>
<td>&quot;cRAND&quot; - adjusted Rand index (default); &quot;RAND&quot; - Rand index</td>
</tr>
<tr>
<td></td>
<td>additional argument passed to <code>dist.SDA</code> function</td>
</tr>
</tbody>
</table>

Details

For HINoV in symbolic data analysis there can be used methods based on distance matrix such as hierarchical ("single", "ward", "complete", "average", "mcquitty", "median", "centroid") and optimization methods ("pam", "DClust") and also methods based on symbolic data table ("SClust").

See file `../doc/HINoVSDA_details.pdf` for further details.

Value

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>parim</td>
<td>$m \times m$ symmetric matrix ($m$ - number of variables). Matrix contains pairwise adjusted Rand (or Rand) indices for partitions formed by the $j$-th variable with partitions formed by the $l$-th variable</td>
</tr>
<tr>
<td>topri</td>
<td>sum of rows of <code>parim</code></td>
</tr>
<tr>
<td>stopri</td>
<td>ranked values of <code>topri</code> in decreasing order</td>
</tr>
</tbody>
</table>

Author(s)

Andrzej Dudek <andrzej.dudek@ue.wroc.pl>, Justyna Wilk <justyna.wilk@ue.wroc.pl> Department of Econometrics and Computer Science, Wroclaw University of Economics, Poland [http://keii.ue.wroc.pl/symbolicDA](http://keii.ue.wroc.pl/symbolicDA)

References


Ichino's feature selection method for symbolic data

Description

Ichino’s method for identifying non-noisy variables in symbolic data set

Usage

IchinoFS.SDA(table.Symbolic)

Arguments

table.Symbolic  symbolic data table

Details

See file ../doc/IchinoFSSDA_details.pdf for further details

Value

plot  plot of the gradient illustrating combinations of variables, in which the axis of ordinates (Y) represents the maximum number of mutual neighbor pairs and the axis of the abscissae (X) corresponds to the number of features (m)

combination  the best combination of variables, i.e. the combination most differentiating the set of objects

maximum results  step-by-step combinations of variables up to m variables

calculation results  .............
Author(s)

Andrzej Dudek <andrzej.dudek@ue.wroc.pl>, Justyna Wilk <justyna.wilk@ue.wroc.pl> Department of Econometrics and Computer Science, Wroclaw University of Economics, Poland http://keii.ue.wroc.pl/symbolicDA

References


See Also

HINoV.SDA; HINoV.Symbolic in clusterSim library

Examples

```r
# LONG RUNNING - UNCOMMENT TO RUN
#data("cars",package="symbolicDA")
#sdt<-cars
#ichino<-IchinoFS.SDA(sdt)
#print(ichino)
```

index.G1d Calinski-Harabasz pseudo F-statistic based on distance matrix

Description

Calculates Calinski-Harabasz pseudo F-statistic based on distance matrix

Usage

```r
index.G1d (d,cl)
```

Arguments

- `d` distance matrix (see `dist.SDA`)
- `cl` a vector of integers indicating the cluster to which each object is allocated

Details

See file `../doc/indexG1d_details.pdf` for further details
Value
value of Calinski-Harabasz pseudo F-statistic based on distance matrix

Author(s)
Andrzej Dudek <andrzej.dudek@ue.wroc.pl>, Justyna Wilk <justyna.wilk@ue.wroc.pl> Department of Econometrics and Computer Science, Wroclaw University of Economics, Poland
http://keii.ue.wroc.pl/symbolicDA

References
Milligan, G.W., Cooper, M.C. (1985), *An Examination of Procedures of Determining the Number of Clusters in a Data Set*, "Psychometrika", Vol. 50, No. 2, pp. 159-179.

See Also

Examples

```r
# LONG RUNNING - UNCOMMENT TO RUN
# Example 1
#library(stats)
data("cars",package="symbolicDA")
x<-cars
d<-dist.SDA(x, type="U_2")
wynik<-hclust(d, method="ward", members=NULL)
clusters<-cutree(wynik, 4)
G1d<-index.G1d(d, clusters)
print(G1d)

# Example 2

data("cars",package="symbolicDA")
md <- dist.SDA(cars, type="U_3", gamma=0.5, power=2)
# nc - number_of_clusters
min_nc=2
max_nc=10
res <- array(0,c(max_nc-min_nc+1,2))
```
# interscal.SDA

```
#res[1,] <- min_nc:max_nc
#clusters <- NULL
#for (nc in min_nc:max_nc)
#{
#cl2 <- pam(md, nc, diss=TRUE)
#res[nc-min_nc+1,2] <- Gl <- index.Glid(md,cl2$clustering)
#clusters <- rbind(clusters, cl2$clustering)
#}
#paste("max Gl for",(min_nc:max_nc)[which.max(res[,2])],"clustering for max Gl")
#write.table(res,file="Gld_res.csv",sep="",dec="",row.names=TRUE,col.names=FALSE)
#plot(res, type="p", pch=0, xlab="Number of clusters", ylab="Gl", xaxt="n")
#axis(1, c(min_nc:max_nc))
```

interscal.SDA

*Multidimensional scaling for symbolic interval data - InterScal algorithm*

**Description**

Multidimensional scaling for symbolic interval data - InterScal algorithm

**Usage**

interscal.SDA(x,d=2,calculateDist=FALSE)

**Arguments**

- `x` symbolic interval data: a 3-dimensional table, first dimension represents object number, second dimension - variable number, and third dimension contains lower- and upper-bounds of intervals (Simple form of symbolic data table)
- `d` Dimensionality of reduced space
- `calculateDist` if TRUE x are treated as raw data and min-max dist matrix is calculated. See details

**Details**

Interscal is the adaptation of well-known classical multidimensional scaling for symbolic data. The input for Interscal is the interval-valued dissimilarity matrix. Such dissimilarity matrix can be obtained from symbolic data matrix (that contains only interval-valued variables), judgements obtained from experts, respondents. See Lechevallier Y. (2001) for details on calculating interval-valued distance. See file `../doc/Symbolic_MDS.pdf` for further details
iscal.SDA

Value

xprim coordinates of rectangles
stress.sym final STRESSSym value

Author(s)

Andrzej Dudek <andrzej.dudek@ue.wroc.pl> Marcin Pelka <marcin.pelka@ue.wroc.pl>
Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland
http://keii.ue.wroc.pl/symbolicDA

References


See Also

iscal.SDA,symscal.SDA

Examples

# LONG RUNNING - UNCOMMENT TO RUN
#sda<-parse.SQ("samochody")
data<-sda$indivIC
#mds<-interscal.SDA(data, d=2, calculateDist=TRUE)

iscal.SDA Multidimensional scaling for symbolic interval data - IScal algorithm

Description

Multidimensional scaling for symbolic interval data - IScal algorithm

Usage

iscal.SDA(x,d=2,calculateDist=FALSE)
Arguments

x symbolic interval data: a 3-dimensional table, first dimension represents object number, second dimension - variable number, and third dimension contains lower- and upper-bounds of intervals (Simple form of symbolic data table)
d Dimensionality of reduced space
calculateDist if TRUE x are treated as raw data and min-max dist matrix is calculated. See details

Details

IScal, which was proposed by Groenen et. al. (2006), is an adaptation of well-known nonmetric multidimensional scaling for symbolic data. It is an iterative algorithm that uses I-STRESS objective function. This function is normalized within the range [0; 1] and can be interpreted like classical STRESS values. IScal, like Interscal and SymScal, requires interval-valued dissimilarity matrix. Such dissmilarity matrix can be obtained from symbolic data matrix (that contains only interval-valued variables), judgements obtained from experts, respondents. See Lechevallier Y. (2001) for details on calculating interval-valued distance. See file ../doc/Symbolic_MDS.pdf for further details

Value

xprim coordinates of rectangles
STRESSSym final STRESSSym value

Author(s)

Andrzej Dudek <andrzej.dudek@ue.wroc.pl>
Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland
http://keii.ue.wroc.pl/symbolicDA

References


See Also

interscal.SDA,symscal.SDA
Examples

# Example will be available in next version of package, thank You for your patience :-)
References


See Also
dist.SDA

Examples

# Example 1
# LONG RUNNING - UNCOMMENT TO RUN
# sda<parse.SO("samochody")
# model<-kernel.SDA(sda, "Typ_samochodu-.", testSet=6:16, h=0.75)
# model

ekohonen.SDA Kohonen’s self-organizing maps for symbolic interval-valued data

Description

Kohonen’s self-organizing maps for a set of symbolic objects described by interval-valued variables

Usage

kohonen.SDA(data, rlen=100, alpha=c(0.05,0.01))

Arguments

data symbolic data table in simple form (see S02Simple)
rlen number of iterations (the number of times the complete data set will be presented to the network)
alpha learning rate, determining the size of the adjustments during training. Default is to decline linearly from 0.05 to 0.01 over rlen updates

Details

See file ../doc/kohonenSDA_details.pdf for further details
Description
Kohonen self organizing maps for symbolic data with interval variables

Usage
parse.SO(file)

Arguments
file file name without xml extension

Details
see symbolic.object for symbolic data table R structure representation
Value
Symbolic data table parsed from XML file

Author(s)
Andrzej Dudek <andrzej.dudek@ue.wroc.pl>
Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland
http://keii.ue.wroc.pl/clusterSim

References

See Also
save.S0, generate.S0

Examples
#cars<-parse.S0("cars")

---

PCA.centers.SDA  principal component analysis for symbolic objects described by symbolic interavl variables. Centers algorithm

Description
principal component analysis for symbolic objects described by symbolic interavl variables. Centers algorithm

Usage
PCA.centers.SDA(t, pc.number=2)

Arguments
t  symbolic interval data: a 3-dimensional table, first dimension represents object number, second dimension - variable number, and third dimension contains lower- and upper-bounds of intervals (Simple form of symbolic data table)

pc.number  number of principal components
Details

See file ../doc/PCA_SDA.pdf for further details

Value

Data in reduced space (symbolic interval data: a 3-dimensional table)

Author(s)

Andrzej Dudek <andrzej.dudek@ue.wroc.pl>

Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland

http://keii.ue.wroc.pl/symbolicDA

References


See Also

PCA.mrprca.SDA, PCA.spaghetti.SDA, PCA.spca.SDA, PCA.vertices.SDA

Examples

# Example will be available in next version of package, thank You for your patience :-)

---

**PCA.mrprca.SDA**

principal component analysis for symbolic objects described by symbolic interval variables. Midpoints and radii algorithm

Description

principal component analysis for symbolic objects described by symbolic interval variables. *Midpoints and radii algorithm*

Usage

```
PCA.mrprca.SDA(t, pc.number=2)
```
Arguments

t  symbolic interval data: a 3-dimensional table, first dimension represents object number, second dimension - variable number, and third dimension contains lower- and upper-bounds of intervals (Simple form of symbolic data table)

pc.number  number of principal components

Details

See file ..//doc/PCA_SDA.pdf for further details

Value

Data in reduced space (symbolic interval data: a 3-dimensional table)

Author(s)

Andrzej Dudek <andrzej.dudek@ue.wroc.pl>

Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland

http://keii.ue.wroc.pl/symbolicDA

References


See Also

PCA.centers.SDA, PCA.spaghetti.SDA, PCA.spca.SDA, PCA.vertices.SDA

Examples

# Example will be available in next version of package, thank You for your patience :-)

PCA.spaghetti.SDA  principal component analysis for symbolic objects described by symbolic interavl variables. Spaghetti algorithm

Description

principal component analysis for symbolic objects described by symbolic interavl variables. Spaghetti algorithm
Usage

```r
PCA.spaghetti.SDA(t, pc.number=2)
```

Arguments

- `t`: symbolic interval data: a 3-dimensional table, first dimension represents object number, second dimension - variable number, and third dimension contains lower- and upper-bounds of intervals (Simple form of symbolic data table)
- `pc.number`: number of principal components

Details

See file `../doc/PCA_SDA.pdf` for further details

Value

Data in reduced space (symbolic interval data: a 3-dimensional table)

Author(s)

Andrzej Dudek <andrzej.dudek@ue.wroc.pl>

Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland
http://keii.ue.wroc.pl/symbolicDA

References


See Also

`PCA.centers.SDA`, `PCA.mrpca.SDA`, `PCA.spca.SDA`, `PCA.vertices.SDA`

Examples

```r
# Example will be available in next version of package, thank You for your patience :-)```

PCA.spca.SDA  

**principal component analysis for symbolic objects described by symbolic interval variables. \textquote{Symbolic}' PCA algorithm**

**Description**

principal component analysis for symbolic objects described by symbolic interval variables. \textquote{Symbolic}' PCA algorithm

**Usage**

PCA.spca.SDA(t, pc.number=2)

**Arguments**

- **t**  
  symbolic interval data: a 3-dimensional table, first dimension represents object number, second dimension - variable number, and third dimension contains lower- and upper-bounds of intervals (Simple form of symbolic data table)
- **pc.number**  
  number of principal components

**Details**

See file 

`./doc/PCA_SDA.pdf` for further details

**Value**

Data in reduced space (symbolic interval data: a 3-dimensional table)

**Author(s)**

Andrzej Dudek <andrzej.dudek@ue.wroc.pl>

Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland

[http://keii.ue.wroc.pl/symbolicDA](http://keii.ue.wroc.pl/symbolicDA)

**References**


**See Also**

PCA.centers.SDA, PCA.mrPCA.SDA, PCA.spaghetti.SDA, PCA.vertices.SDA
Examples

# Example will be available in next version of package, thank You for your patience :-)
plot3dInterval

3D plot for symbolic interval-valued data

Description

3-dimensional plot for symbolic objects described by interval-valued variables

Usage

plot3dInterval(data, colors)

Arguments

data symbolic data table consists of a set of symbolic objects described by interval-valued variables
colors set of colors (see colors) to mark symbolic objects

Value

3-dimensional plot for symbolic interval-valued data in which each axis represents a symbolic interval-valued variable and each cuboid represents a symbolic object

Author(s)

Andrzej Dudek <andrzej.dudek@ue.wroc.pl>, Justyna Wilk <justyna.wilk@ue.wroc.pl> Department of Econometrics and Computer Science, Wroclaw University of Economics, Poland http://keii.ue.wroc.pl/symbolicDA

References


See Also

PCA.centers.SDA, PCA.mrpca.SDA, PCA.spaghetti.SDA, PCA.spca.SDA

Examples

# Example will be available in next version of package, thank You for your patience :-)

plot3dInterval(data, colors)
Examples

```R
# LONG RUNNING - UNCOMMENT TO RUN
#set.seed(1)
#means <- matrix(c(0,0,0,
#                   0,0,6,
#                   0,6,0,
#                   0,6,6,
#                   6,0,0,
#                   6,0,6,
#                   6,6,0,
#                   6,6,6),8,3,byrow=TRUE)
#means <- means*1.5
#means[,] <- means[,] - 2
#cov <- matrix(c(1,0,0,0,1,0,0,0,1),3,3)
#t <- cluster.Gen(model=2, means=means, cov=cov, dataType="s", numObjects=10)
#plot3dInterval(t$data, colors=rainbow(8)[t$clusters])
#rgl.viewpoint(15,20,30)
#rgl.snapshot("8_clusters_3d.jpg")
```

random.forest.SDA  

*Random forest algorithm for optimal split based decision tree for symbolic objects*

Description

Random forest algorithm for optimal split based decision tree for symbolic objects

Usage

`random.forest.SDA(sdt, formula, testSet, mfinal = 100,...)`

Arguments

- `sdt`  Symbolic data table
- `formula`  formula as in `lm` function
- `testSet`  a vector of integers indicating classes to which each objects are allocated in learning set
- `mfinal`  number of partial models generated
- `...`  arguments passed to `decisionTree.SDA` function

Details

`random.forest.SDA` implements Breiman’s random forest algorithm for classification of symbolic data set.

Value

Section details goes here
Author(s)
Andrzej Dudek <andrzej.dudek@ue.wroc.pl> Marcin Pelka <marcin.pelka@ue.wroc.pl>
Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland
http://keii.ue.wroc.pl/symbolicDA

References
Billard L., Diday E. (red.) (2006), Symbolic Data Analysis, Conceptual Statistics and Data Mining,
John Wiley & Sons, Chichester.
statistical information from complex data, Springer-Verlag, Berlin.
Diday E., Noirhomme-Fraiture M. (red.) (2008), Symbolic Data Analysis with SODAS Software,
John Wiley & Sons, Chichester.

See Also
bagging.SDA,boosting.SDA,decisionTree.SDA

Examples
# Example will be available in next version of package, thank You for your patience :-)

replication.SDA  Modification of replication analysis for cluster validation of symbolic
data

Description
Replication analysis for cluster validation of symbolic data

Usage
replication.SDA(table.Symbolic, u=2, method="SClust", S=10, fixedAsample=NULL, ...)

Arguments
table.Symbolic  symbolic data table
u              number of clusters given arbitrarily
method         clustering method: "SClust" (default), "DClust", "single", "complete", "average",
               "mcquitty", "median", "centroid", "ward", "pam", "diana"
S              the number of simulations used to compute average adjusted Rand index
fixedAsample   if NULL A sample is generated randomly, otherwise this parameter contains
               object numbers arbitrarily assigned to A sample
...            additional argument passed to dist.SDA function
Details

See file ../doc/repliationSDA_details.pdf for further details

Value

A 3-dimensional array containing data matrices for A sample of objects in each simulation (first dimension represents simulation number, second - object number, third - variable number)

B 3-dimensional array containing data matrices for B sample of objects in each simulation (first dimension represents simulation number, second - object number, third - variable number)

medoids 3-dimensional array containing matrices of observations on u representative objects (medoids) for A sample of objects in each simulation (first dimension represents simulation number, second - cluster number, third - variable number)

clusteringA 2-dimensional array containing cluster numbers for A sample of objects in each simulation (first dimension represents simulation number, second - object number)

clusteringB 2-dimensional array containing cluster numbers for B sample of objects in each simulation (first dimension represents simulation number, second - object number)

clusteringBB 2-dimensional array containing cluster numbers for B sample of objects in each simulation according to 4 step of replication analysis procedure (first dimension represents simulation number, second - object number)

cRand value of average adjusted Rand index for S simulations

Author(s)

Andrzej Dudek <andrzej.dudek@ue.wroc.pl>, Justyna Wilk <justyna.wilk@ue.wroc.pl> Department of Econometrics and Computer Science, Wroclaw University of Economics, Poland http://keii.ue.wroc.pl/symbolicDA

References


Diday E., Noirhomme-Fraiture M. (Eds.) (2008), Symbolic Data Analysis with SODAS Software, John Wiley & Sons, Chichester.
See Also

dist.SDA, SClust, DClust: hclust in stats library; pam in cluster library; replication.Mod in clustersim library

Examples

```r
#data("cars",package="symbolicDA")
#set.seed(123)
#w<-repliation.SDA(cars, u=3, method="SClust", S=10)
#print(w)
```

```r
save.SO(sdt, file) saves symbolic data table of 'symbolic' class to xml file
```

Description

saves symbolic data table of 'symbolic' class to xml file (ASSO format)

Usage

```r
save.SO(sdt, file)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdt</td>
<td>Symbolic data table</td>
</tr>
<tr>
<td>file</td>
<td>file name with extension</td>
</tr>
</tbody>
</table>

Details

see `symbolic.object` for symbolic data table R structure representation

Value

No value returned

Author(s)

Andrzej Dudek <andrzej.dudek@ue.wroc.pl>
Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland
http://keii.ue.wroc.pl/symbolicDA

References

Dynamical clustering of symbolic data

Usage

SClust(table.Symbolic, cl, iter=100, variableSelection=NULL, objectSelection=NULL)

Arguments

table.Symbolic  symbolic data table
cl              number of clusters or vector with initial prototypes of clusters
iter            maximum number of iterations
variableSelection vector of numbers of variables to use in clustering procedure or NULL for all variables
objectSelection  vector of numbers of objects to use in clustering procedure or NULL for all objects

Details

See file ../doc/SClust_details.pdf for further details

Value

a vector of integers indicating the cluster to which each object is allocated

Author(s)

Andrzej Dudek <andrezj.dudek@ue.wroc.pl>, Justyna Wilk <justyna.wilk@ue.wroc.pl> Department of Econometrics and Computer Science, Wroclaw University of Economics, Poland http://keii.ue.wroc.pl/symbolicDA
References


See Also

`DClust; kmeans` in stats library

Examples

```r
# LONG RUNNING - UNCOMMENT TO RUN
#data("cars", package="symbolicDA")
#sdt<-cars
#clust<-SClust(sdt, cl=3, iter=50)
#print(clust)
```

**simple2SO**

Change of representation of symbolic data from simple form to symbolic data table

Description

Change of representation of symbolic data from simple form to symbolic data table

Usage

```r
simple2SO(x)
```

Arguments

- `x` symbolic interval data: a 3-dimensional table, first dimension represents object number, second dimension - variable number, and third dimension contains lower- and upper-bounds of intervals

Details

see `symbolic.object` for symbolic data table R structure representation
Value
Symbolic data table in full form

Author(s)
Andrzej Dudek <andrzej.dudek@ue.wroc.pl>
Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland
http://keii.ue.wroc.pl/symbolicDA

References

See Also
link{SO2Simple}

Examples

# Example will be available in next version of package, thank You for your patience :-)

SO2Simple

| SO2Simple | Change of representation of symbolic data from symbolic data table to simple form |

Description
Change of representation of symbolic data from symbolic data table to simple form

Usage
SO2Simple(sd)

Arguments
sd Symbolic data table in full form

Details
see *symbolic.object* for symbolic data table R structure representation
**Value**

symbolic interval data: a 3-dimensional table, first dimension represents object number, second dimension - variable number, and third dimension contains lower- and upper-bounds of intervals

**Author(s)**

Andrzej Dudek <andrzej.dudek@ue.wroc.pl>
Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland
http://keii.ue.wroc.pl/symbolicDA

**References**


**See Also**

link{simple2S0}

**Examples**

# Example will be available in next version of package, thank You for your patience :-)

```r

subsdts.DA sdt objectselection

**Description**

This method creates symbolic data table containing only objects, whose indices are given in secng argument

**Usage**

subsdts.DA(sdt,objectSelection)

**Arguments**

sdt Symbolic data table
objectSelection vector containing symbolic object numbers, default value - all objects from sdt
symbolic.object

Details

see symbolic.object for symbolic data table R structure representation

Value

Symbolic data table containing only objects, whose indices are given in second argument. The result is of 'symbolic' class

Author(s)

Andrzej Dudek <andrzej.dudek@ue.wroc.pl>

Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland

http://keii.ue.wroc.pl/symbolicDA

References


See Also

generate.SO, save.SO, parse.SO

Examples

# Example will be available in next version of package, thank You for your patience :-)
Value

**individuals** data frame with one row for each row in symbolic data table with following columns:
- `num` - symbolic object (described by symbolic data table row) ordering number, usually from 1 to number of symbolic objects;
- `name` - short name of symbolic object with no spaces;
- `label` - full descriptive name of symbolic object.

**variables** data frame with one row for each column in symbolic data table with following columns:
- `num` - symbolic variable (adequate to symbolic data table column) ordering number, usually from 1 to number of symbolic variables;
- `name` - short name of symbolic variable with no spaces;
- `label` - full descriptive name of symbolic variable;
- `type` - type of symbolic variable: `IC` (InterContinous) - Symbolic interval variable type, every realization of symbolic variable of this type on symbolic object takes form of numerical interval; `C` (Continous) - Symbolic interval variable type, every realization of symbolic variable of this type on symbolic object takes form of numerical interval for which begging is equal to end (equivalent to simple "numeric" value); `MN` (MultiNominal) - every realization of multi nominal symbolic variable on symbolic objects takes form of set of nominal values; `NM` (Multi Nominal Modif) - every realization of nominal symbolic variable on symbolic objects takes form of distribution of probabilities (set of nominal values with weights summing to one); `N` (Nominal) - every realization of nominal symbolic variable on symbolic objects is one value (or N.A.)
- `details` - id of this variable in details table appropriate for this kind of variable (`detailsN` for nominal and multi nominal variables, `detailsIC` for symbolic interval variables, `detailsC` for continous (metric single-valued) variables, `detailsNM` of multi nominal with weights variables).

**detailsC** data frame describing symbolic continous (metric, single-valued) variables details with following columns:
- `na` - number of N.A. (not available) variables realization;
- `nu` - not used, left for compatibility with ASSO-XML specification;
- `min` - beginning of interval representing symbolic interval variable domain (minimal value of all realizations of this variable on all symbolic objects);
- `max` - end of interval representing symbolic interval variable domain (maximal value of all realizations of this variable on all symbolic objects).

**detailsIC** data frame describing symbolic inter-continous (symbolic interval) variables details with following columns:
- `na` - number of N.A. (not available) variables realizations;
- `nu` - not used, left for compatibility with ASSO-XML specification;
- `min` - beginning of interval representing symbolic interval variable domain (minimal value of all beginnings of interval realizations of this variable on all symbolic objects);
- `max` - end of interval representing symbolic interval variable domain (maximal value of all ends of interval realizations of this variable on all symbolic objects).
symbolic.object

**detailsN** data frame describing symbolic nominal and multi nominal variables details with following columns:
- `na` - number of N.A. variables realizations;
- `nu` - not used, left for compatibility with ASSO-XML specification;
- `modals` - number of categories in symbolic variable domain. Each categorie is described in `detailsListNom`.

**detailsListNom** data frame describing every category of symbolic nominal and multi nominal variables, with following columns:
- `details_no` - number of variable in `detailsN` to which domain belongs category;
- `num` - number of category within variable domain;
- `name` - category short name
- `label` - category full name

**detailsNM** data frame describing symbolic multi nominal modiff (categories sets with weights) variables details with following columns:
- `na` number of N.A. (not available) variables realizations.
- `nu` not used, left for compatibility with ASSO-XML specification
- `modals` number of categories in symbolic variable domain. Each categorie is described in `detailsListNomModiff`.

**detailsListNomModiff** data frame describing every category of symbolic multi nominal modiff variables, with following columns:
- `details_no` - number of variable in `detailsNM` to which domain belongs category;
- `num` - number of category within variable domain;
- `name` - category short name
- `label` - category full name

**indivIC** array of symbolic interval variables realizations, with dimensions `nr_of_objects X nr_of_variables X 2` containing beginnings and ends of intervals for given object and variable. For values different type than symbolic interval array contains zeros.

**indivC** array of symbolic continues variables realizations, with dimensions `nr_of_objects X nr_of_variables X 1` containing single values - realizations of variable on symbolic object. For values different type than symbolic continous array contains zeros.

**indivN** data frame describing symbolic nominal and multi nonimal variables realizations with folowing columns:
- `indiv` - id of symbolic object from `individuals`;
- `variable` - id of symbolic object from `variables`;
- `value` - id of category object from `detailsListNom`;

When this data frame contains line `i,j,k` it means that category `k` belongs to set that is realization of `j`-th symbolic variable on `i`-th symbolic object.

**indivNM** data frame describing symbolic multi nonimal modiff variables realizations with folowing columns:
- `indiv` - id of symbolic object from `individuals`;

variable - id of symbolic object from variables;
value - id of category object from detailsListNom;
frequency - weight of category;

When this data frame contains line $i,j,k,w$ it means that category $k$ belongs to set that is realization of $j$-th symbolic variable on $i$-th symbolic object with weight(probability) $w$.

Structure

The following components must be included in a legitimate symbolic object.

See Also

dist.SDA.

---

**symscal.SDA**  
*Multidimensional scaling for symbolic interval data - SymScal algorithm*

**Description**

Multidimensional scaling for symbolic interval data - SymScal algorithm

**Usage**

`symscal.SDA(x,d=2,calculateDist=FALSE)`

**Arguments**

- `x` symbolic interval data: a 3-dimensional table, first dimension represents object number, second dimension - variable number, and third dimension contains lower- and upper-bounds of intervals (Simple form of symbolic data table)
- `d` Dimensionality of reduced space
- `calculateDist` if TRUE x are treated as raw data and min-max dist matrix is calculated. See details

**Details**

SymScal, which was proposed by Groenen et. al. (2005), is an adaptation of well-known non-metric multidimensional scaling for symbolic data. It is an iterative algorithm that uses STRESS objective function. This function is unnormalized. IScal, like Interscal and SymScal, requires interval-valued dissimilarity matrix. Such dissimilarity matrix can be obtained from symbolic data matrix (that contains only interval-valued variables), judgements obtained from experts, respondents. See Lechevallier Y. (2001) for details on calculating interval-valued distance. See file `../doc/Symbolic_MDS.pdf` for further details.
Value

xprim coordinates of rectangles
STRESSSym final STRESSSym value

Author(s)

Andrzej Dudek <andrzej.dudek@ue.wroc.pl>
Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland
http://keii.ue.wroc.pl/symbolicDA

References


See Also

iscal.SDA, interscal.SDA

Examples

# Example will be available in next version of package, thank You for your patience :-)

```
```
Arguments

- `table.Symbolic` symbolic data table
- `j` symbolic object number in symbolic data table used to create the chart
- `variableSelection` numbers of symbolic variables describing symbolic object used to create the chart, if NULL all variables are used
- `offset` relational offset of chart (margin size)
- `firstTick` place of first tick (relational to length of axis)
- `labelCex` labels cex parameter of labels
- `labelOffset` relational offset of labels
- `tickLength` relational length of single tick of axis
- `histWidth` histogram (for modal variables) relational width
- `histHeight` histogram (for modal variables) relational height
- `rotateLabels` if TRUE labels are rotated due to rotation of axes
- `variableCex` cex parameter of names of variables

Value

zoom star chart for selected symbolic object in which each axis represents a symbolic variable. Depending on the type of symbolic variable their implementations are presented as:

a) rectangle - interval range of interval-valued variable,
b) circles - categories of multinominal (or multinominal with weights) variable from among coloured circles means categories of the variable observed for the selected symbolic object
bar chart - additional chart for multinominal with weights variable in which each bar represents a weight (percentage share) of a category of the variable

Author(s)

Andrzej Dudek <andrzej.dudek@ue.wroc.pl>, Justyna Wilk <justyna.wilk@ue.wroc.pl> Department of Econometrics and Computer Science, Wroclaw University of Economics, Poland http://keii.ue.wroc.pl/symbolicDA

References


See Also

`plot3dInterval`, `plotInterval` in clusterSim
Examples

# LONG RUNNING - UNCOMMENT TO RUN
# Example 1
#data("cars", package="symbolicDA")
#sd<-cars
#zoomStar(sdt, j=12)

# Example 2
#data("cars", package="symbolicDA")
#sd<-cars
#variables<-as.matrix(sdt$variables)
#indivN<-as.matrix(sdt$indivN)
#dist<-as.matrix(dist.SDA(sdt))
#classes<-DClust(dist, cl=5, iter=100)
#for(i in 1:max(classes)){
#  #getOption("device")()
#  #zoomStar(sdt, .medoid(dist, classes, i))}
Index

*Topic MDS
- interscal.SDA, 20
- iscal.SDA, 21
- symscal.SDA, 44

*Topic Multidimensional
- interscal.SDA, 20
- iscal.SDA, 21
- symscal.SDA, 44

*Topic PCA
- PCA.centers.SDA, 26
- PCA.mrPCA.SDA, 27
- PCA.spaghetti.SDA, 28
- PCA.sPCA.SDA, 30
- PCA.vertices.SDA, 31

*Topic Principal components analysis
- PCA.centers.SDA, 26
- PCA.mrPCA.SDA, 27
- PCA.spaghetti.SDA, 28
- PCA.sPCA.SDA, 30
- PCA.vertices.SDA, 31

*Topic SDA
- symbolic.object, 41

*Topic cluster
- bagging.SDA, 2
- boosting.SDA, 4
- cluster.Description.SDA, 6
- DClust, 8
- HIHoV.SDA, 15
- IchinoFS.SDA, 17
- index.G1d, 18
- kernel.SDA, 23
- kohonen.SDA, 24
- parse.SO, 25
- replication.SDA, 34
- save.SO, 36
- SClust, 37
- simple2SO, 38
- SO2Simple, 39
- subsdt.SDA, 40

*Topic datasets
- cars, 5
- data_symbolic, 8

*Topic decision tree
- draw.decisionTree.SDA, 13
- random_forest.SDA, 33

*Topic hplot
- plot3dInterval, 32

*Topic multivariate
- replication.SDA, 34

*Topic scaling
- interscal.SDA, 20
- iscal.SDA, 21
- symscal.SDA, 44

*Topic symbolic, SDA
- decisionTree.SDA, 9
- dist.SDA, 11
- generate.SO, 14

*Topic symbolic
- symbolic.object, 41

- bagging.SDA, 2, 5, 11, 34
- boosting.SDA, 4, 4, 11, 34
- cars, 5
- cluster.Description.SDA, 6
- colors, 32
- data_symbolic, 8
- DClust, 7, 8, 13, 19, 36, 38
- decisionTree.SDA, 4, 5, 9, 14, 34
- dist.SDA, 9, 11, 16, 18, 24, 34, 36, 44
- draw.decisionTree.SDA, 11, 13

- generate.SO, 14, 26, 37, 41
- hclust, 7
- HIHoV.SDA, 15, 18
- IchinoFS.SDA, 17

zoomStar, 45

48
INDEX

index.G1d, 13, 18
interscal.SDA, 20, 22, 45
iscal.SDA, 21, 21, 45

kernel.SDA, 23
kmeans, 38
kohonen.SDA, 24

parse.SO, 25, 37, 41
PCA.centers.SDA, 26, 28–30, 32
PCA.mrPCA.SDA, 27, 27, 29, 30, 32
PCA.spaghetti.SDA, 27, 28, 28, 30, 32
PCA.sPCA.SDA, 27–29, 30, 32
PCA.vertices.SDA, 27–30, 31
plot3dInterval, 32, 46

random.forest.SDA, 4, 5, 11, 33
replication.SDA, 34

save.SO, 26, 36, 41
SClust, 7, 9, 19, 32, 36, 37
simple2SO, 38
SO2Simple, 24, 39
subsdt.SDA, 37, 40
symbolic.object, 15, 25, 36, 38, 39, 41, 41
symscal.SDA, 21, 22, 44

zoomStar, 45