The mvpart Package

November 16, 2005

Version 1.1-1

Date November 2004 version of mvpart

Author rpart by Terry M Therneau and Beth Atkinson <atkinson@mayo.edu>. R port of rpart by Brian Ripley <ripley@stats.ox.ac.uk>. Some routines from vegan – Jari Oksanen <jari.oksanen@oulu.fi> Extensions and adaptations of rpart to mvpart by Glenn De’ath <g.death@aims.gov.au>.

Maintainer Glenn De’ath <g.death@aims.gov.au>

Description Multivariate regression trees

Title Multivariate partitioning

Depends R (>= 2.0.0)

Suggests survival

License use under GPL2, or see file LICENCE

R topics documented:

car.test.frame ................................................................. 2
cmds.diss ................................................................. 3
descendants ................................................................. 4
distfull ................................................................. 4
eqscpt ................................................................. 5
formatg ................................................................. 5
gdist ................................................................. 5
kyphosis ................................................................. 7
labels.rpart .......................................................... 8
meanvar.rpart .......................................................... 9
model.frame.rpart .................................................... 10
mvpart ............................................................... 10
na.rpart ............................................................... 11
node.match ........................................................... 12
path.rpart ............................................................ 12
plot.rpart ............................................................ 13
plotcp ............................................................... 14
post.rpart ............................................................ 15
predict.rpart .......................................................... 17
### Description

The `car.test.frame` data frame has 60 rows and 8 columns, giving data on makes of cars taken from the April, 1990 issue of *Consumer Reports*.

### Usage

```r
data(car.test.frame)
```
cmds.diss

Format
This data frame contains the following columns:

- **Price**: a numeric vector giving the list price in US dollars of a standard model.
- **Country**: of origin, a factor with levels France Germany Japan Japan/USA Korea Mexico Sweden USA.
- **Reliability**: a numeric vector coded 1 to 5.
- **Mileage**: fuel consumption miles per US gallon, as tested.
- **Type**: a factor with levels Compact Large Medium Small Sporty Van.
- **Weight**: kerb weight in pounds.
- **HP**: the engine capacity (displacement) in litres.

Examples
```
data(car.test.frame)
z.auto <- rpart(Mileage ~ Weight, car.test.frame)summary(z.auto)
```

**cmds.diss**

**Classical Scaling of Dissimilarity Measures**

Description
The function first computes the dissimilarity matrix according to the specified method – see gdist or xdiss. The dissimilarities are then scaled using classical scaling – see cmdscale. The returned matrix can be input into rpart or mvpart for multivariate regression tree splitting.

Usage
```
cmds.diss(data, k = ncol(data), x.use = FALSE, method = "man", zero.chk = TRUE, plt = FALSE, plot.subset = FALSE, plot.subn = 5)
```

Arguments
```
data                Data matrix
k                   Number of vectors to be returned
x.use               Use extended dissimilarity?
method              Dissimilarity index
zero.chk            Check for zero row sums – if zero ignore these rows according to method
plt                 Plot the relationship between the dissimilarities and the distances calculated from the scaled output vectors.
```
**plot.subset**  Plot a subset of the points – useful for large data sets.

**plot.subn**  Controls how many points are plotted when plot.subset=TRUE. The number of points plotted is 750 + N * plot.subn where N = number of rows in data.

**Details**

The function knows the same dissimilarity indices as `gdist`. Plotting the relationship between the dissimilarities and the distances calculated from the scaled output vectors is useful in assessing potential loss of information. If the loss is high then the results from partitioning directly from the dissimilarity matrix using distance-base partitioning (see `dist` in `rpart`), and those obtained from partitioning the output of `cmds.diss` using multivariate regression trees (see `mrt` in `rpart`) can be substantial.

**Author(s)**

Glenn De’ath

**Examples**

```r
data(spider)
dist.vecs <- cmds.diss(spider)

# comparing splitting using "dist" and "mrt" methods
# for euclidean distance the answers are indentical :
# first using "mrt" on the data directly
mvpart(data.matrix(spider[,1:12])~water+twigs+reft+herbs+moss+sand,spider,method="mrt",size=5)

# now using "dist" -- note we need the full distance matrix squared
mvpart(gdist(spider[,1:12],meth="euc",full=TRUE,sq=TRUE)~water+twigs+reft+herbs+moss+sand,spider,method="dist",size=5)

# finally using "mrt" from the scaled dissimilarities.
mvpart(cmds.diss(spider[,1:12],meth="euc")~water+twigs+reft+herbs+moss+sand,spider,method="mrt",size=5)

# try with some other measure of dissimilarity eg extended bray-curtis -- the result will differ
# between methods
```

---

**descendants**  Descendants

**Description**

Internal function for Rpart

**Details**

descendants is to be used only internally.
<table>
<thead>
<tr>
<th>distfull</th>
<th>Distfull</th>
</tr>
</thead>
</table>

**Description**

Internal function for Rpart

**Details**

distfull is to be used only internally.

<table>
<thead>
<tr>
<th>eqscplt</th>
<th>Eqscplt</th>
</tr>
</thead>
</table>

**Description**

Internal function for Rpart

**Details**
eqscplt is to be used only internally.

<table>
<thead>
<tr>
<th>formatg</th>
<th>Formatg</th>
</tr>
</thead>
</table>

**Description**

Internal function for Rpart

**Details**

formatg is to be used only internally.
**gdist**

**Dissimilarity Measures**

**Description**

The function computes useful dissimilarity indices which are known to have a good rank-order relation with gradient separation and are thus efficient in community ordination with multidimensional scaling.

**Usage**

```r
gdist(x, method="bray", keepdiag=FALSE, full=FALSE, sq=FALSE)
```

**Arguments**

- `x` Data matrix
- `method` Dissimilarity index
- `keepdiag` Compute and keep diagonals
- `full` Return the square dissimilarity matrix
- `sq` Square the dissimilarities – useful for distance-based partitioning

**Details**

The function knows the following dissimilarity indices:

- **euclidean**  
  \[ d_{jk} = \sqrt{\sum_i (x_{ij} - x_{ik})^2} \]
- **manhattan**  
  \[ d_{jk} = \sum_i |x_{ij} - x_{ik}| \]
- **gower**  
  \[ d_{jk} = \sum_i \frac{|x_{ij} - x_{ik}|}{\max_i - \min_i} \]
- **canberra**  
  \[ d_{jk} = \frac{1}{N-Z} \sum_i \frac{|x_{ij} - x_{ik}|}{x_{ij} + x_{ik}} \]
- **bray**  
  \[ d_{jk} = \frac{\sum_i (x_{ij} + x_{ik})}{\sum_i (x_{ij} + x_{ik})} \]
- **kulczynski**  
  \[ d_{jk} = 1 - 0.5\left(\frac{\sum \min(x_{ij}, x_{ik})}{\sum x_{ij}} + \frac{\sum \min(x_{ij}, x_{ik})}{\sum x_{ik}}\right) \]
- **maximum**  
  \[ d_{jk} = \max_i |x_{ij} - x_{ik}| \]
- **binary**  
  \[ d_{jk} = \sum_i |x_{ij} > 0 - x_{ik} > 0| \]
- **chord**  
  \[ d_{jk} = \sqrt{\sum_i (x_{ij} - x_{ik})^2 / \sum_i (x_{ij} + x_{ik})^2} \]

where \(N - Z\) is the number of non-zero entries.

Infamous "double zeros" are removed in Canberra dissimilarity.

Euclidean and Manhattan dissimilarities are not good in gradient separation without proper standardization but are still included for comparison and special needs.

Some of indices become identical or rank-order similar after some standardizations.

**Value**

Should be interchangeable with `dist` and returns a distance object of the same type.
Note

The function is an alternative to dist adding some ecologically meaningful indices. Both methods should produce similar types of objects which can be interchanged in any method accepting either. Manhattan and Euclidean dissimilarities should be identical in both methods, and Canberra dissimilarity may be similar.

Author(s)

Jari Oksanen – modified Glenn De’ath (Dec 03)

References


Examples

data(spider)
spider.dist <- gdist(spider[1:12,])

data(kyphosis)

Description

The kyphosis data frame has 81 rows and 4 columns. representing data on children who have had corrective spinal surgery

Usage

data(kyphosis)

Format

This data frame contains the following columns:

- **Kyphosis** a factor with levels absent present indicating if a kyphosis (a type of deformation) was present after the operation.
- **Age** in months
- **Number** the number of vertebrae involved
- **Start** the number of the first (topmost) vertebra operated on.

Source

Examples

```r
data(kyphosis)
fit <- rpart(Kyphosis ~ Age + Number + Start, data=kyphosis)
fit2 <- rpart(Kyphosis ~ Age + Number + Start, data=kyphosis,
              parms=list(prior=c(.65,.35), split='information'))
fit3 <- rpart(Kyphosis ~ Age + Number + Start, data=kyphosis,
              control=rpart.control(cp=.05))
par(mfrow=c(1,2))
plot(fit)
text(fit, use.n=TRUE)
plot(fit2)
text(fit2, use.n=TRUE)
```

labels.rpart  

Create Split Labels For an Rpart Object

Description

This function provides labels for the branches of an rpart tree.

Usage

```r
## S3 method for class 'rpart':
labels(object, digits=4, minlength=1, pretty, collapse=TRUE, ...)
```

Arguments

- `object` fitted model object of class rpart. This is assumed to be the result of some function that produces an object with the same named components as that returned by the rpart function.
- `digits` the number of digits to be used for numeric values. All of the rpart functions that call labels explicitly set this value, with `options("digits")` as the default.
- `minlength` the minimum length for abbreviation of character or factor variables. If 0 no abbreviation is done; if 1 then single letters are used with "a" for the first level, "b" for the second and so on. If the value is greater than 1, the abbreviate function is used.
- `pretty` an argument included for backwards compatibility: pretty=0 implies `minlength`=0, pretty=NULL implies `minlength`=1, and pretty=TRUE implies `minlength`=4.
- `collapse` logical. The returned set of labels is always of the same length as the number of nodes in the tree. If `collapse`=TRUE (default), the returned value is a vector of labels for the branch leading into each node, with "root" as the label for the top node. If FALSE, the returned value is a two column matrix of labels for the left and right branches leading out from each node, with "leaf" as the branch labels for terminal nodes.
- `...` optional arguments to abbreviate.

Value

Vector of split labels (collapse=TRUE) or matrix of left and right splits (collapse=FALSE) for the supplied rpart object. This function is called by printing methods for rpart and is not intended to be called directly by the users.
See Also

abbreviate

meanvar.rpart               Mean-Variance Plot for an Rpart Object

Description

Creates a plot on the current graphics device of the deviance of the node divided by the number of
observations at the node. Also returns the node number.

Usage

## S3 method for class 'rpart':
meanvar(tree, xlab="ave(y)", ylab="ave(deviance)", ...)

Arguments

tree      fitted model object of class rpart. This is assumed to be the result of some func-
tion that produces an object with the same named components as that returned
by the rpart function.

xlab      x-axis label for the plot.

ylab      y-axis label for the plot.

...       additional graphical parameters may be supplied as arguments to this function.

Value

an invisible list containing the following vectors is returned.

x          fitted value at terminal nodes (yval).

y          deviance of node divided by number of observations at node.

label      node number.

Side Effects

a plot is put on the current graphics device.

See Also

plot.rpart.

Examples

data(car.test.frame)
z.auto <- rpart(Mileage ~ Weight, car.test.frame)
meanvar(z.auto, log='xy')
model.frame.rpart  Model.frame.rpart

Description

Internal function for Rpart

Details

model.frame.rpart is to be used only internally.

mvpart  Recursive Partitioning and Regression Trees

Description

Wrapper function for fitting and plotting rpart models

Usage

mvpart(form, data, minauto = TRUE, size, xv = c("1se", "min", "pick", "none"), xval = 10, xvmult = 0, xvse = 1, snip = FALSE, plot.add = TRUE, text.add = TRUE, digits = 3, margin = 0, uniform = FALSE, which = 4, pretty = TRUE, use.n = TRUE, all.leaves = FALSE, bars = TRUE, legend, bord = FALSE, xadj = 1, yadj = 1, prn = FALSE, branch = 1, rsq = FALSE, big.pts = FALSE, pca = FALSE, interact.pca = FALSE, wgt.ave.pca = FALSE, ...)

Arguments

form  As for rpart function. Arguments to rpart can be passed by ...

data  Optional data frame in which to interpret the variables named in the formula

minauto  If TRUE uses smart minsplit and minbucket based on N cases.

size  The size of tree to be generated.

xv  Selection of tree by cross-validation: "1se" - gives best tree within one SE of the overall best, "min" - the best tree, "pick" - pick the tree size interactively, "none" - no cross-validation.

xval  Number of cross-validations or vector defining cross-validation groups.

xvmult  Number of multiple cross-validations.

xvse  Multiplier for the number of SEs used for xv = "1se".

plot.add  Plot the tree and (optionally) add text.

text.add  Add output of text.rpart to tree.

snip  Interactively prune the tree.

digits  Number of digits on labels.

margin  Margin around plot, 0.1 gives an extra 10 percent space around the plot.
uniform Uniform lengths to the branches of the tree.
which Which split labels and where to plot them, 1=centered, 2 = left, 3 = right and 4 = both.
pretty Pretty labels or full labels.
use.n Add number of cases at each node.
all.leaves Annotate all nodes.
bars If TRUE adds barplots to nodes.
legend If TRUE adds legend for mrt and classification trees.
bord Border (box) around the barplots.
xadj, yadj Adjust the size of the individual barplots (default = 1).
prn If TRUE prints tree details.
branch Controls spread of branches: 1=vertical lines, 0=maximum slope.
rsq If TRUE gives "rsq" plot.
big.pts Plot colored points at leaves – useful to link to PCA plot.
pca If TRUE plots PCA of group means and add species and site information.
interact.pca If TRUE runs interactive PCA. See rpart.pca.
wgt.ave.pca If TRUE plot weighted averages across sites for species.
... ...other arguments passed to rpart.

Value

an object of class rpart, a superset of class tree.

See Also

rpart, rpart.pca.

Examples

data(spider)
mvpart(data.matrix(spider[,1:12])~herbs+reft+moss+sand+twigs+water,spider) # defaults
mvpart(data.matrix(spider[,1:12])~herbs+reft+moss+sand+twigs+water,spider,xv="p") # pick cv size and do PCA
fit <- mvpart(data.matrix(spider[,1:12])~herbs+reft+moss+sand+twigs+water,spider,xv="1se")
rpart.pca(fit,interact=TRUE,wgt.ave=TRUE) # interactive PCA plot of saved multivariate tree

na.rpart

Handles Missing Values in an Rpart Object

Description

Handles missing values in an rpart object.

Usage

na.rpart(x)
Arguments

\( x \) a model frame.

Details

Internal function that handles missing values when calling the function \( \text{rpart} \).

\[ \text{node.match} \] \( \text{Node.match} \)

Description

Internal function for \( \text{Rpart} \)

Details

\( \text{node.match} \) is to be used only internally.

\[ \text{path.rpart} \] \( \text{Follow Paths to Selected Nodes of an Rpart Object} \)

Description

Returns a names list where each element contains the splits on the path from the root to the selected nodes.

Usage

\[ \text{path.rpart}(\text{tree}, \text{nodes}, \text{pretty}=0, \text{print.it}=\text{TRUE}) \]

Arguments

tree fitted model object of class \( \text{rpart} \). This is assumed to be the result of some function that produces an object with the same named components as that returned by the \( \text{rpart} \) function.

nodes an integer vector containing indices (node numbers) of all nodes for which paths are desired. If missing, user selects nodes as described below.

pretty an integer denoting the extent to which factor levels in split labels will be abbreviated. A value of (0) signifies no abbreviation. A \( \text{NULL} \), the default, signifies using elements of letters to represent the different factor levels.

print.it Logical. Denotes whether paths will be printed out as nodes are interactively selected. Irrelevant if \( \text{nodes} \) argument is supplied.

Details

The function has a required argument as an \( \text{rpart} \) object and a list of nodes as optional arguments. Omitting a list of nodes will cause the function to wait for the user to select nodes from the dendrogram. It will return a list, with one component for each node specified or selected. The component contains the sequence of splits leading to that node. In the graphical interaction, the individual paths are printed out as nodes are selected.
Value

A named (by node) list, each element of which contains all the splits on the path from the root to the specified or selected nodes.

Graphical Interaction

A dendrogram of the \texttt{rpart} object is expected to be visible on the graphics device, and a graphics input device (e.g., a mouse) is required. Clicking (the selection button) on a node selects that node. This process may be repeated any number of times. Clicking the exit button will stop the selection process and return the list of paths.

References

This function was modified from \texttt{path.tree} in S.

See Also

\texttt{rpart}

Examples

\begin{verbatim}
data(kyphosis)
fit <- rpart(Kyphosis ~ Age + Number + Start, data=kyphosis)
summary(fit)
path.rpart(fit, node=c(11, 22))
\end{verbatim}

---

\textbf{plot \texttt{.rpart}} \hspace{1cm} \textit{Plot an Rpart Object}

Description

Plots an \texttt{rpart} object on the current graphics device.

Usage

\begin{verbatim}
## S3 method for class 'rpart':
plot(x, uniform=FALSE, branch=1, compress=FALSE, nspace, margin=0, minbranch=.3, bar, ms.fudge = FALSE,...)
\end{verbatim}

Arguments

- \texttt{x} \hspace{1cm} a fitted object of class \texttt{rpart}, containing a classification, regression, or rate tree.
- \texttt{uniform} \hspace{1cm} if TRUE, uniform vertical spacing of the nodes is used; this may be less cluttered when fitting a large plot onto a page. The default is to use a non-uniform spacing proportional to the error in the fit.
- \texttt{branch} \hspace{1cm} controls the shape of the branches from parent to child node. Any number from 0 to 1 is allowed. A value of 1 gives square shouldered branches, a value of 0 give V shaped branches, with other values being intermediate.
### Details

This function is a method for the generic function `plot`, for objects of class `rpart`. The y-coordinate of the top node of the tree will always be 1.

### Value

the coordinates of the nodes are returned as a list, with components `x` and `y`.

### Side Effects

an unlabeled plot is produced on the current graphics device.

### See Also

- `rpart`
- `text.rpart`

### Examples

```r
data(car.test.frame)
fit <- rpart(Price ~ Mileage + Type + Country, car.test.frame)
plot(fit, compress=TRUE)
text(fit, use.n=TRUE)
```

---

**plotcp**

Plot a Complexity Parameter Table for an Rpart Fit

**Description**

Gives a visual representation of the cross-validation results in an `rpart` object.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>compress</strong></td>
<td>If <code>FALSE</code>, the leaf nodes will be at the horizontal plot coordinates of <code>1:nleaves</code>. If <code>TRUE</code>, the routine attempts a more compact arrangement of the tree. The compaction algorithm assumes <code>uniform=TRUE</code>; surprisingly, the result is usually an improvement even when that is not the case.</td>
</tr>
<tr>
<td><strong>nspace</strong></td>
<td>the amount of extra space between a node with children and a leaf, as compared to the minimal space between leaves. Applies to compressed trees only. The default is the value of <code>branch</code>.</td>
</tr>
<tr>
<td><strong>margin</strong></td>
<td>an extra percentage of white space to leave around the borders of the tree. (Long labels sometimes get cut off by the default computation).</td>
</tr>
<tr>
<td><strong>minbranch</strong></td>
<td>set the minimum length for a branch to <code>minbranch</code> times the average branch length. This parameter is ignored if <code>uniform=TRUE</code>. Sometimes a split will give very little improvement, or even (in the classification case) no improvement at all. A tree with branch lengths strictly proportional to improvement leaves no room to squeeze in node labels.</td>
</tr>
<tr>
<td><strong>bar</strong></td>
<td>length of bar at root (default = 0.03) – used instead of char &quot;</td>
</tr>
<tr>
<td><strong>ms.fudge</strong></td>
<td>If <code>FALSE</code> adjust &quot;legs&quot; of plots to avoid MicroSoft metafile stuff-ups!!</td>
</tr>
<tr>
<td><strong>...</strong></td>
<td>arguments to be passed to or from other methods.</td>
</tr>
</tbody>
</table>
**Usage**

```r
cp = 2
plotcp(x, xvse = 1, minline = TRUE, upper = c("size", "splits", "none"), tab, resub.err = TRUE, adj.df = FALSE, ...)
```

**Arguments**

- `x`: an object of class `rpart`
- `xvse`: multiplier for `xvse * SE` above the minimum of the curve.
- `minline`: whether a horizontal line is drawn 1SE above the minimum of the curve.
- `upper`: what is plotted on the top axis: the size of the tree (the number of leaves), the number of splits or nothing.
- `tab`: used for multiple cross-validation.
- `resub.err`: use resubstitution error for calculations of SEs.
- `adj.df`: adjust df of resubstitution error estimate for calculations of SEs.
- `...`: additional plotting parameters

**Details**

The set of possible cost-complexity prunings of a tree from a nested set. For the geometric means of the intervals of values of `cp` for which a pruning is optimal, a cross-validation has (usually) been done in the initial construction by `rpart`. The `cptable` in the fit contains the mean and standard deviation of the errors in the cross-validated prediction against each of the geometric means, and these are plotted by this function. A good choice of `cp` for pruning is often the leftmost value for which the mean lies below the horizontal line.

**Value**

None.

**Side Effects**

A plot is produced on the current graphical device.

**See Also**

- `rpart.printcp`
- `rpart.object`

---

**Description**

Generates a PostScript presentation plot of an `rpart` object.

**Usage**

```r
## S3 method for class 'rpart':
post(tree, title.,
     filename = paste(deparse(substitute(tree)), ".ps", sep = ""),
     digits = getOption("digits") - 3, pretty = TRUE,
     use.n = TRUE, horizontal = TRUE, ...)
```
Arguments

tree
fitted model object of class \texttt{rpart}. This is assumed to be the result of some function that produces an object with the same named components as that returned by the \texttt{rpart} function.

title.
a title which appears at the top of the plot. By default, the name of the \texttt{rpart} endpoint is printed out.

filename
ASCII file to contain the output. By default, the name of the file is the name of the object given by \texttt{rpart} (with the suffix .\texttt{ps} added). If \texttt{filename = ""}, the plot appears on the current graphical device.

digits
number of significant digits to include in numerical data.

pretty
an integer denoting the extent to which factor levels will be abbreviated in the character strings defining the splits; (0) signifies no abbreviation of levels. A NULL signifies using elements of letters to represent the different factor levels. The default (\texttt{TRUE}) indicates the maximum possible abbreviation.

use.n
Logical. If \texttt{TRUE} (default), adds to label (#events level1/ #events level2/etc. for method \texttt{class}, n for method \texttt{anova}, and #events/n for methods \texttt{poisson} and \texttt{exp}).

horizontal
Logical. If \texttt{TRUE} (default), plot is horizontal. If \texttt{FALSE}, plot appears as landscape.

... other arguments to the \texttt{postscript} function.

Details

The plot created uses the functions \texttt{plot.rpart} and \texttt{text.rpart} (with the \texttt{fancy} option). The settings were chosen because they looked good to us, but other options may be better, depending on the \texttt{rpart} object. Users are encouraged to write their own function containing favorite options.

Side Effects

a plot of \texttt{rpart} is created using the \texttt{postscript} driver, or the current device if \texttt{filename = ""}.

See Also

\texttt{plot.rpart}, \texttt{rpart}, \texttt{text.rpart}, \texttt{abbreviate}

Examples

data(car.test.frame)
z.auto <- \texttt{rpart(Mileage ~ Weight, car.test.frame)}
post(z.auto, file = "")  # display tree on active device
  # now construct postscript version on file "pretty.ps"
  # with no title
post(z.auto, file = "pretty.ps", title = " ")
z.hp <- \texttt{rpart(Mileage ~ Weight + HP, car.test.frame)}
post(z.hp)
predict.rpart

Predictions from a Fitted Rpart Object

Description
Returns a vector of predicted responses from a fitted rpart object.

Usage

## S3 method for class 'rpart':
predict(object, newdata=list(),
    type=c("vector", "prob", "class", "matrix"), ...)

Arguments

object fitted model object of class rpart. This is assumed to be the result of some function that produces an object with the same named components as that returned by the rpart function.

newdata data frame containing the values at which predictions are required. The predictors referred to in the right side of formula(object) must be present by name in newdata. If missing, the fitted values are returned.

type character string denoting the type of predicted value returned. If the rpart object is a classification tree, then the default is to return prob predictions, a matrix whose columns are the probability of the first, second, etc. class. (This agrees with the default behavior of tree). Otherwise, a vector result is returned.

... further arguments passed to or from other methods.

Details
This function is a method for the generic function predict for class rpart. It can be invoked by calling predict for an object of the appropriate class, or directly by calling predict.rpart regardless of the class of the object.

Value
A new object is obtained by dropping newdata down the object. For factor predictors, if an observation contains a level not used to grow the tree, it is left at the deepest possible node and frame$yval at the node is the prediction.

If type="vector":
vector of predicted responses. For regression trees this is the mean response at the node, for Poisson trees it is the estimated response rate, and for classification trees it is the predicted class.

If type="prob":
(for a classification tree) a matrix of class probabilities.

If type="matrix":
a matrix of the full responses (frame$yval2 if this exists, otherwise frame$yval). For regression trees, this is the mean response, for Poisson trees it is the response rate and the number of events at that node in the fitted tree, and for classification trees it is the concatenation of the predicted class, the class counts at that node in the fitted tree, and the class probabilities.

If type="class":
(for a classification tree) a factor of classifications based on the responses.
print.rpart

See Also

predict.rpart.object

Examples

data(car.test.frame)
z.auto <- rpart(Mileage ~ Weight, car.test.frame)
predict(z.auto)

data(kyphosis)
fit <- rpart(Kyphosis ~ Age + Number + Start, data=kyphosis)
predict(fit, type="prob") # class probabilities (default)
predict(fit, type="vector") # level numbers
predict(fit, type="class") # factor
predict(fit, type="matrix") # level number, class frequencies, probabilities

data(iris)
sub <- c(sample(1:50, 25), sample(51:100, 25), sample(101:150, 25))
fit <- rpart(Species ~ ., data=iris, subset=sub)
fit
table(predict(fit, iris[-sub,], type="class"), iris[-sub, "Species"])

print.rpart

Print an Rpart Object

Description

This function prints an rpart object. It is a method for the generic function print of class rpart.

Usage

## S3 method for class 'rpart':
print(x, minlength=0, spaces=2, cp, digits=getOption("digits"), ...)

Arguments

x fitted model object of class rpart. This is assumed to be the result of some function that produces an object with the same named components as that returned by the rpart function.

minlength Controls the abbreviation of labels: see labels.rpart.

spaces the number of spaces to indent nodes of increasing depth.

digits the number of digits of numbers to print.

cp prune all nodes with a complexity less than cp from the printout. Ignored if unspecified.

... arguments to be passed to or from other methods.

Details

This function is a method for the generic function print for class "rpart". It can be invoked by calling print for an object of the appropriate class, or directly by calling print.rpart regardless of the class of the object.
printcp

Side Effects

A semi-graphical layout of the contents of \texttt{x$frame} is printed. Indentation is used to convey the tree topology. Information for each node includes the node number, split, size, deviance, and fitted value. For the \texttt{"class"} method, the class probabilities are also printed.

See Also

\texttt{print.rpart.object}, \texttt{summary.rpart}, \texttt{printcp}

Examples

data(car.test.frame)
z.auto <- rpart(Mileage ~ Weight, car.test.frame)
z.auto

## Not run: node), split, n, deviance, yval
* denotes terminal node

1) root 60 1354.58300 24.58333
2) Weight>=2567.5 45 361.20000 22.46667
   4) Weight>=3087.5 22 61.31818 20.40909 *
   5) Weight<3087.5 23 117.65220 24.43478
      10) Weight>=2747.5 15 60.40000 23.80000 *
      11) Weight<2747.5 8 39.87500 25.62500 *
3) Weight<2567.5 15 186.93330 30.93333 *
## End(Not run)

\texttt{printcp} Displays CP table for Fitted Rpart Object

Description

Displays the \texttt{cp} table for fitted \texttt{rpart} object.

Usage

\texttt{printcp(x, digits=getOption("digits") - 2)}

Arguments

\texttt{x} fitted model object of class \texttt{rpart}. This is assumed to be the result of some function that produces an object with the same named components as that returned by the \texttt{rpart} function.

\texttt{digits} the number of digits of numbers to print.

Details

Prints a table of optimal prunings based on a complexity parameter.

See Also

\texttt{summary.rpart}, \texttt{rpart.object}
Examples

```r
data(car.test.frame)
z.auto <- rpart(Mileage ~ Weight, car.test.frame)
printcp(z.auto)
## Not run:
Regression tree:
rpart(formula = Mileage ~ Weight, data = car.test.frame)

Variables actually used in tree construction:
[1] Weight

Root node error: 1354.6/60 = 22.576

<table>
<thead>
<tr>
<th>CP</th>
<th>nsplit</th>
<th>rel error</th>
<th>xerror</th>
<th>xstd</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>0.00000</td>
<td>1.03436</td>
<td>0.178526</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.40465</td>
<td>0.60508</td>
<td>0.105217</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0.27012</td>
<td>0.45153</td>
<td>0.083330</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0.25729</td>
<td>0.44826</td>
<td>0.076998</td>
</tr>
</tbody>
</table>
## End(Not run)
```

prune.rpart

Cost-complexity Pruning of an Rpart Object

Description

Determines a nested sequence of subtrees of the supplied `rpart` object by recursively snipping off the least important splits, based on the complexity parameter (cp).

Usage

```r
## S3 method for class 'rpart':
prune(tree, cp, ...)
```

Arguments

- `tree` fitted model object of class `rpart`. This is assumed to be the result of some function that produces an object with the same named components as that returned by the `rpart` function.
- `cp` Complexity parameter to which the `rpart` object will be trimmed.
- `...` further arguments passed to or from other methods.

Value

A new `rpart` object that is trimmed to the value `cp`.

See Also

`rpart`
**Examples**

```r
data(car.test.frame)
z.auto <- rpart(Mileage ~ Weight, car.test.frame)
zp <- prune(z.auto, cp=0.1)
plot(zp) #plot smaller rpart object
```

---

**residuals.rpart**  
Residuals From a Fitted Rpart Object

**Description**

Method for residuals for an `rpart` object.

**Usage**

```r
## S3 method for class 'rpart':
residuals(object, type = c("usual", "pearson", "deviance"), ...)
```

**Arguments**

- `object`  
  fitted model object of class "rpart".

- `type`  
  Indicates the type of residual desired.
  - For regression or anova trees all three residual definitions reduce to \( y - \text{fitted} \). This is the residual returned for user method trees as well.
  - For classification trees the usual residuals are the misclassification losses \( L(\text{actual}, \text{predicted}) \) where \( L \) is the loss matrix. With default losses this residual is 0/1 for correct/incorrect classification. The pearson residual is \( (1 - \text{fitted})/\sqrt{\text{fitted}(1-\text{fitted})} \) and the deviance residual is \( \sqrt{-2\log(\text{fitted})} \).
  - For poisson and exp (or survival) trees, the usual residual is the observed - expected number of events. The pearson and deviance residuals are as defined in McCullagh and Nelder.
  - ...

- `...`  
  further arguments passed to or from other methods.

**Value**

vector of residuals of type `type` from a fitted `rpart` object.

**References**


**Examples**

```r
data(solder)
fit <- rpart(skips ~ Opening + Solder + Mask + PadType + Panel,
             data=solder, method='anova')
summary(residuals(fit))
plot(predict(fit), residuals(fit))
```
Description
Fit a rpart model

Usage
rpart(formula, data=NULL, weights, subset, na.action=na.rpart, method,
model=FALSE, x=FALSE, y=TRUE, parms, control, cost, ...)

Arguments
formula a formula, as in the lm function.
data an optional data frame in which to interpret the variables named in the formula
weights optional case weights.
subset optional expression saying that only a subset of the rows of the data should be used in the fit.
na.action The default action deletes all observations for which y is missing, but keeps those in which one or more predictors are missing.
method one of "anova", "poisson", "class", "mrt", "dist", or "exp". If method is missing then the routine tries to make an intelligent guess. If y is a survival object, then method="exp" is assumed, if y is a matrix then method="mrt" is assumed, if y is a factor then method="class" is assumed, otherwise method="anova" is assumed. It is wisest to specify the method directly, especially as more criteria are added to the function. For method="dist" the response must be a square symmetric distance matrix; e.g. returned by gdist or xdiss. Weights and cross-validation are currently not implemented for method="dist". Alternatively, method can be a list of functions named init, split and eval.
model keep a copy of the model frame in the result. If the input value for model is a model frame (likely from an earlier call to the rpart function), then this frame is used rather than constructing new data.
x keep a copy of the x matrix in the result.
y keep a copy of the dependent variable in the result.
parms optional parameters for the splitting function. Anova splitting has no parameters. Poisson splitting has a single parameter, the coefficient of variation of the prior distribution on the rates. The default value is 1. Exponential splitting has the same parameter as Poisson. For classification splitting, the list can contain any of: the vector of prior probabilities (component prior), the loss matrix (component loss) or the splitting index (component split). The priors must be positive and sum to 1. The loss matrix must have zeros on the diagonal and positive off-diagonal elements. The splitting index can be gini or information. The default priors are proportional to the data counts, the losses default to 1, and the split defaults to gini.
control options that control details of the rpart algorithm.
cost  
a vector of non-negative costs, one for each variable in the model. Defaults to one for all variables. These are scalings to be applied when considering splits, so the improvement on splitting on a variable is divided by its cost in deciding which split to choose.

Arguments to rpart.control may also be specified in the call to rpart. They are checked against the list of valid arguments.

Details

This differs from the tree function mainly in its handling of surrogate variables. In most details it follows Breiman et. al. quite closely.

Value

an object of class rpart, a superset of class tree.

References


See Also

rpart.control, rpart.object, summary.rpart, print.rpart

Examples

data(car.test.frame)
z.auto <- rpart(Mileage ~ Weight, car.test.frame)
summary(z.auto)
plot(z.auto); text(z.auto)
data(spider)
fit1 <- rpart(data.matrix(spider[,1:12])~water+twigs+reft+herbs+moss+sand,spider,method="*
fit2 <- rpart(gdist(spider[,1:12],meth="bray",full=TRUE,sq=TRUE)~water+twigs+reft+herbs+m
par(mfrow=c(1,2))
plot(fit1); text(fit1)
plot(fit2); text(fit2)

rpart.anova  Set up for ‘anova’ Method for an Rpart Model

Description

Set up for anova method for an rpart model.

Usage

rpart.anova(y, offset, parms, wt)
Arguments

- **y**: the responses.
- **offset**: an offset, or `NULL`.
- **parms**: a list of parameters, usually empty.
- **wt**: case weights.

Details

`rpart.anova` is to be used only by the function `rpart`.

Value

Returns a list consisting of

- **y**: (subtracting offset if necessary).
- **parms**: as input,
- **numresp**: the number of responses, here 1.
- **summary**: a function to be invoked by `summary.rpart`.
- **text**: a function to be invoked by `text.rpart`.

---

**rpart.branch**

*Rpart.branch*

Description

Internal function for Rpart

Details

`rpart.branch` is to be used only internally.

---

**rpart.class**

Set up for `class` Method for an Rpart Model

Description

Set up for the `class` method for an `rpart` model.

Usage

`rpart.class(y, offset, parms, wt)`

Arguments

- **y**: the responses.
- **offset**: `NULL`: anything else is an error.
- **parms**: a named list of parameters.
- **wt**: case weights.
Details

rpart.class is to be used only by the function rpart. It validates the input parameter list: if that is missing or NULL default values are supplied.

Value

Returns a list consisting of

- **y**: `y` as factor codes.
- **parms**: A named list of parameters. This will have components `prior`, `loss` and `split` (and more if supplied).
- **numresp**: The number of responses (one plus the number of classes).
- **counts**: A vector of counts for each class.
- **ylevels**: The response levels corresponding to the codes given in `y`.
- **summary**: A function to be invoked by `summary.rpart`.
- **text**: A function to be invoked by `text.rpart`.

See Also

- `rpart`

rpart.control  
*Control for Rpart Models*

Description

Various parameters that control aspects of the `rpart` fit.

Usage

```r
rpart.control(minsplit=5, minbucket=round(minsplit/3), cp=0.01,
               maxcompete=4, maxsurrogate=5, usesurrogate=2, xval=10,
               surrogatestyle=0, maxdepth=30, ...)
```

Arguments

- **minsplit**: The minimum number of observations that must exist in a node, in order for a split to be attempted.
- **minbucket**: The minimum number of observations in any terminal `<leaf>` node. If only one of `minbucket` or `minsplit` is specified, the code either sets `minsplit` to `minbucket*3` or `minbucket` to `minsplit/3`, as appropriate.
- **cp**: Complexity parameter. Any split that does not decrease the overall lack of fit by a factor of `cp` is not attempted. For instance, with `anova` splitting, this means that the overall Rsquare must increase by `cp` at each step. The main role of this parameter is to save computing time by pruning off splits that are obviously not worthwhile. Essentially, the user informs the program that any split which does not improve the fit by `cp` will likely be pruned off by cross-validation, and that hence the program need not pursue it.
maxcompete  the number of competitor splits retained in the output. It is useful to know not just which split was chosen, but which variable came in second, third, etc.

maxsurrogate  the number of surrogate splits retained in the output. If this is set to zero the compute time will be shortened, since approximately half of the computational time (other than setup) is used in the search for surrogate splits.

usesurrogate  how to use surrogates in the splitting process. 0= display only; an observation with a missing value for the primary split rule is not sent further down the tree. 1= use surrogates, in order, to split subjects missing the primary variable; if all surrogates are missing the observation is not split. 2= if all surrogates are missing, then send the observation in the majority direction. A value of 0 corresponds to the action of tree, and 2 to the recommendations of Breiman, et.al.

xval  number of cross-validations

surrogatestyle  controls the selection of a best surrogate. If set to 0 (default) the program uses the total number of correct classification for a potential surrogate variable, if set to 1 it uses the percent correct, calculated over the non-missing values of the surrogate. The first option more severely penalizes covariates with a large number of missing values.

maxdepth  Set the maximum depth of any node of the final tree, with the root node counted as depth 0 (past 30 rpart will give nonsense results on 32-bit machines).

...  mop up other arguments.

Value

a list containing the options.

See Also

rpart

rpart.dist  Set up for 'dist' Method for an Rpart Model

Description

Set up for dist method for an rpart model.

Usage

rpart.dist(y, offset, parms, wt)

Arguments

y  the responses.

offset  an offset, or NULL.

parms  a list of parameters, usually empty.

wt  case weights – ignored.
Details

`rpart.exp` is to be used only by the function `rpart`.

Value

Returns a list consisting of

- `y` (subtracting `offset` if necessary).
- `parms` as input,
- `numresp` the number of responses,
- `summary` a function to be invoked by `summary.rpart`,
- `text` a function to be invoked by `text.rpart`.

Description

Set up for the `exp` method for an `rpart` model.

Usage

```r
rpart.exp(y, offset, parms, wt)
```

Arguments

- `y` the responses, a vector or a two-column matrix.
- `offset` an offset, or `NULL`.
- `parms` a list of parameters.
- `wt` case weights.

Details

`rpart.exp` is to be used only by the function `rpart`. It transforms the data to look like a unit-rate Poisson process so that the `rpart.exp` method can be used.

Value

Returns a list consisting of

- `y` a matrix giving the observation times and number of events,
- `parms` a named list of parameters, with components `shrink` and `method` (1 for "deviance", 2 for "sqrt"),
- `numresp` the number of responses, here 2,
- `numy` the number of columns in `y`, here 2,
- `summary` a function to be invoked by `summary.rpart`,
- `text` a function to be invoked by `text.rpart`. 
rpart.matrix  

*Creates a Model Matrix from a Call to Rpart*

**Description**

Creates a model matrix from a call to `rpart`.

**Usage**

```r
rpart.matrix(frame)
```

**Arguments**

- `frame`  
  model frame (from call to `rpart`)

**Value**

Returns a matrix from the frame of class `matrix`. Used internally by the function `rpart`.

---

rpart.mrt  

*Set up for 'mrt' Method for an Rpart Model*

**Description**

Set up for `mrt` method for an `rpart` model.

**Usage**

```r
rpart.mrt(y, offset, parms, wt)
```

**Arguments**

- `y`  
  the responses.
- `offset`  
  an offset, or `NULL`.
- `parms`  
  a list of parameters, usually empty.
- `wt`  
  case weights.

**Details**

`rpart.mrt` is to be used only by the function `rpart`.

**Value**

Returns a list consisting of

- `y` (subtracting `offset` if necessary),
- `parms` as input,
- `numresp` the number of responses,
- `summary` a function to be invoked by `summary.rpart`.
- `text` a function to be invoked by `text.rpart`.
- `bar` a function to be invoked by `text.rpart`.
rpart.object

Recursive Partitioning and Regression Trees Object

**Description**

These are objects representing fitted rpart trees.

**Value**

- **frame**
  - data frame with one row for each node in the tree. The *row.names* of *frame* contain the (unique) node numbers that follow a binary ordering indexed by node depth. Elements of *frame* include *var*, the variable used in the split at each node (leaf nodes are denoted by the string *<leaf>*), *n*, the size of each node, *wt*, the sum of case weights for the node, *dev*, the deviance of each node, *yval*, the fitted value of the response at each node, and *splits*, a two column matrix of left and right split labels for each node. All of these are the same as for an *rpart* object.
  
  Extra response information is in *yval2*, which contains the number of events at the node (poisson), or a matrix containing the fitted class, the class counts for each node and the class probabilities (classification). Also included in the frame are *complexity*, the complexity parameter at which this split will collapse, *ncompete*, the number of competitor splits retained, and *nsurrogate*, the number of surrogate splits retained.

- **where**
  - vector, the same length as the number of observations in the root node, containing the row number of *frame* corresponding to the leaf node that each observation falls into.

- **splits**
  - a matrix describing the splits. The row label is the name of the split variable, and columns are *count*, the number of observations sent left or right by the split (for competitor splits this is the number that would have been sent left or right had this split been used, for surrogate splits it is the number missing the primary split variable which were decided using this surrogate), *ncat*, the number of categories or levels for the variable (+/-1 for a continuous variable), *improve*, which is the improvement in deviance given by this split, or, for surrogates, the concordance of the surrogate with the primary, and *split*, the numeric split point. The last column *adj* gives the adjusted concordance for surrogate splits. For a factor, the *split* column contains the row number of the csplit matrix. For a continuous variable, the sign of *ncat* determines whether the subset *x < cutpoint* or *x > cutpoint* is sent to the left.

- **csplit**
  - this will be present only if one of the split variables is a factor. There is one row for each such split, and column *i* = -1 if this level of the factor goes to the left, +1 if it goes to the right, and 0 if that level is not present at this node of the tree. For an ordered categorical variable all levels are marked as R/L, including levels that are not present.

- **method**
  - the method used to grow the tree.

- **cptable**
  - the table of optimal prunings based on a complexity parameter.

- **terms**
  - an object of mode *expression* and class *term* summarizing the formula. Used by various methods, but typically not of direct relevance to users.
call
an image of the call that produced the object, but with the arguments all named and with the actual formula included as the formula argument. To re-evaluate the call, say \texttt{update(tree)}.
Optional components include the matrix of predictors (\(x\)) and the response variable (\(y\)) used to construct the \texttt{rpart} object.

Structure
The following components must be included in a legitimate \texttt{rpart} object. Of these, only the \texttt{where} component has the same length as the data used to fit the \texttt{rpart} object.

See Also
\texttt{rpart}.

\texttt{rpart.pca}  
\textit{Principle Components Plot of a Multivariate Rpart Object}

Description
Plots a PCA of the \texttt{rpart} object on the current graphics device.

Usage

\begin{verbatim}
## S3 method for class 'pca':
rpart(tree, pts = TRUE, plt.allx = TRUE, speclab = TRUE,
      specvecs = TRUE, wgt.ave = FALSE, add.tree = TRUE,
      cv1 = 1, cv2 = 2, interact = FALSE, ...)
\end{verbatim}

Arguments

\begin{itemize}
  \item \texttt{tree} A fitted object of class \texttt{rpart} containing a multivariate regression tree.
  \item \texttt{pts} If \texttt{TRUE}, large points representing the leaf means are plotted.
  \item \texttt{plt.allx} If \texttt{TRUE}, small points representing individual cases are plotted.
  \item \texttt{speclab} If \texttt{TRUE} the labels of the response variables are plotted.
  \item \texttt{specvecs} If \texttt{TRUE} the vectors of the response variables are plotted provided \texttt{wgt.ave} is \texttt{FALSE}.
  \item \texttt{wgt.ave} If \texttt{TRUE} use weighted averages of responses not vectors.
  \item \texttt{add.tree} If \texttt{TRUE} add the tree structure to the plot.
  \item \texttt{cv1} Defines the principal component to plot horizontally – but see \texttt{interact}.
  \item \texttt{cv2} Defines the principal component to plot vertically – but see \texttt{interact}.
  \item \texttt{interact} If \texttt{TRUE} the plot can be viewed in dimensions by left-clicking to top-left, bottom-right or bottom-left (reset).
  \item \texttt{...} arguments to be passed to or from other methods.
\end{itemize}

Details
This function plots a PCA biplot of the group means (leaves) of multivariate regression objects of class \texttt{rpart}. The responses and group means and individual cases can be shown on the plot. If responses are positive (e.g., species-environment data) weighted averages of responses can be plotted.
**Value**

`NULL`

**Side Effects**

A PCA biplot plot is produced on the current graphics device.

**See Also**

`rpart`, `text.rpart`

**Examples**

```r
data(spider)
fit<-mvpart(data.matrix(spider[,1:12])~herbs+ref+mos+sand+twigs+water,spider)
rpart.pca(fit)
rpart.pca(fit,wgt.ave=TRUE,interact=TRUE)
```

---

**rpart.poisson**  
*Set up for ‘poisson’ Method for an Rpart Model*

**Description**

Set up for poisson method for an rpart model.

**Usage**

```r
rpart.poisson(y, offset, parms, wt)
```

**Arguments**

- `y`  
  - the responses, a vector or a two-column matrix. If a matrix the first column is the observation times and the second is the counts.

- `offset`  
  - an offset, or `NULL`. The offset is interpreted as a log observation time.

- `parms`  
  - a list of parameters.

- `wt`  
  - case weights.

**Details**

`rpart.poisson` is to be used only by the function `rpart`.

**Value**

Returns a list consisting of

- `y`  
  - a matrix giving the observation times and number of events,

- `parms`  
  - a named list of parameters, with components `shrink` and `method` (1 for "deviance", 2 for "sqrt"),

- `numresp`  
  - the number of responses, here 2,

- `numy`  
  - the number of columns in `y`, here 2,

- `summary`  
  - a function to be invoked by `summary.rpart`,

- `text`  
  - a function to be invoked by `text.rpart`. 
rpconvert

Description

Internal function for Rpart

Details

rpartco is to be used only internally.

rpconvert

Update an rpart object

Description

Rpart objects changed (slightly) in their internal format in order to accommodate the changes for user-written split functions. This routine updates an old object to the new format.

Usage

rpconvert(x)

Arguments

x an rpart object

Value

an updated object

See Also

rpart
rsq.rpart

Plots the Approximate R-Square for the Different Splits

Description

Produces 2 plots. The first plots the r-square (apparent and apparent - from cross-validation) versus the number of splits. The second plots the Relative Error(cross-validation) +/- 1-SE from cross-validation versus the number of splits.

Usage

rsq.rpart(x)

Arguments

x fitted model object of class rpart. This is assumed to be the result of some function that produces an object with the same named components as that returned by the rpart function.

Side Effects

Two plots are produced.

Note

The labels are only appropriate for the "anova" method.

Examples

data(car.test.frame)
z.auto <- rpart(Mileage ~ Weight, car.test.frame)
rsq.rpart(z.auto)

scaler

Row and Column Scaling of a Data Matrix

Description

The function provides some popular (and effective) standardization methods for community ecologists.

Usage

scaler(x, col = NULL, row=NULL)

Arguments

x Data matrix.
col Character vector of column standardizations.
row Character vector of row standardizations.
Details

The function offers following data matrix standardizations:

- mean1: scale to mean of 1.
- max1: scale to maximum of 1.
- ssq1: scale to sums of squares equal 1.
- range01: scale range to 0-1.
- zsc: standardize to z-scores (mean=0, sd=1).
- pa: scale to presence/absence scale (0/1).
- rank: scale to rank order (1=lowest).

Standardizations are performed first on columns then on rows. "pa" applies to the whole matrix and can be specified using row or col.

Value

Returns the standardized matrix.

Note

Common transformations can be made with standard R functions.

Author(s)

Jari Oksanen – modified Glenn De’ath (Dec 03)

Examples

data(spider)
spid.data <- scaler(spider, col = "max", row="mean1")

snip.rpart

Snip Subtrees of an Rpart Object

Description

Creates a "snipped" rpart object, containing the nodes that remain after selected subtrees have been snipped off. The user can snip nodes using the toss argument, or interactively by clicking the mouse button on specified nodes within the graphics window.

Usage

snip.rpart(x, toss)

Arguments

x: fitted model object of class rpart. This is assumed to be the result of some function that produces an object with the same named components as that returned by the rpart function.
toss: an integer vector containing indices (node numbers) of all subtrees to be snipped off. If missing, user selects branches to snip off as described below.
snip.rpart.mouse

Details

A dendrogram of \texttt{rpart} is expected to be visible on the graphics device, and a graphics input device (e.g., a mouse) is required. Clicking (the selection button) on a node displays the node number, sample size, response yvalue, and Error (dev). Clicking a second time on the same node snips that subtree off and visually erases the subtree. This process may be repeated an number of times. Warnings result from selecting the root or leaf nodes. Clicking the exit button will stop the snipping process and return the resulting \texttt{rpart} object.

See the documentation for the specific graphics device for details on graphical input techniques.

Value

a \texttt{rpart} object containing the nodes that remain after specified or selected subtrees have been snipped off.

Warning

Visually erasing the plot is done by over-plotting with the background colour. This will do nothing if the background is transparent (often true for screen devices).

See Also

\texttt{plot.rpart}

Examples

```r
## dataset not in R
## Not run:
z.survey <- rpart(market.survey) #grow the rpart object
plot(z.survey) #plot the tree
z.survey2 <- snip.rpart(z.survey,toss=2) #trim subtree at node 2
plot(z.survey2) #plot new tree

# can also interactively select the node using the mouse in the
# graphics window
## End(Not run)
```

snip.rpart.mouse  Snip.rpart.mouse

Description

Internal function for Rpart

Details

\texttt{snip.rpart.mouse} is to be used only internally.
solder  
_Soldering of Components on Printed-Circuit Boards_

**Description**

The **solder** data frame has 720 rows and 6 columns, representing a balanced subset of a designed experiment varying 5 factors on the soldering of components on printed-circuit boards.

**Usage**

```r
data(solder)
```

**Format**

This data frame contains the following columns:

- **Opening** a factor with levels L M S indicating the amount of clearance around the mounting pad.
- **Solder** a factor with levels Thick Thin giving the thickness of the solder used.
- **Mask** a factor with levels A1.5 A3 B3 B6 indicating the type and thickness of mask used.
- **PadType** a factor with levels D4 D6 D7 L4 L6 L7 L8 L9 W4 W9 giving the size and geometry of the mounting pad.
- **Panel** 1:3 indicating the panel on a board being tested.
- **skips** a numeric vector giving the number of visible solder skips.

**Source**


**Examples**

```r
data(solder)
fit <- rpart(skips ~ Opening + Solder + Mask + PadType + Panel,
             data=solder, method='anova')
summary(residuals(fit))
plot(predict(fit), residuals(fit))
```

---

spider  
_Spider Data_

**Description**

The **spider** data frame has 28 rows and 18 columns. The first 12 columns are abundances of different species of spiders and the next 6 are environmental data.

**Usage**

```r
data(spider)
```
Format

This data frame contains the following columns:

-arct.lute, pard.lugu, zora.spin, pard.nigr, pard.pull, aulo.albi, troc.terr, alop.cune, pard.mont, alop.acce, alop.fabr, arct.peri-
numeric vectors giving the abundances of 12 species of spider

-water, sand, moss, reft, twigs, herbs numeric vectors giving the values of 6 environmental characteristics

Source


Examples

data(spider)
fit<-mvpart(as.matrix(spider[,1:12])~water+twigs+reft+herbs+moss+sand,spider)
summary(fit)

string.bounding.box

Description

Internal function for Rpart

Details

string.bounding.box is to be used only internally.

sub.barplot

Description

Internal function for Rpart

Details

sub.barplot is to be used only internally.
### summary.rpart

**Summarize a Fitted rpart Object**

**Description**

Returns a detailed listing of a fitted rpart object.

**Usage**

```r
## S3 method for class 'rpart':
summary(object, cp=0, digits=getOption("digits"), file, ...)
```

**Arguments**

- **object**: fitted model object of class rpart. This is assumed to be the result of some function that produces an object with the same named components as that returned by the rpart function.
- **digits**: Number of significant digits to be used in the result.
- **cp**: trim nodes with a complexity of less than cp from the listing.
- **file**: write the output to a given file name. (Full listings of a tree are often quite long).
- **...**: arguments to be passed to or from other methods.

**Details**

This function is a method for the generic function summary for class "rpart". It can be invoked by calling summary for an object of the appropriate class, or directly by calling summary.rpart regardless of the class of the object.

**See Also**

`summary`, `rpart.object`, `printcp`.

**Examples**

```r
data(car.test.frame)
z.auto <- rpart(Mileage ~ Weight, car.test.frame)
summary(z.auto)
```

### text.rpart

**Place Text on a Dendrogram**

**Description**

Labels the current plot of the tree dendrogram with text.
## S3 method for class 'rpart':
text(x, splits = TRUE, which = 4, label = "yval", FUN = text,
    all.leaves = FALSE, pretty = NULL, digits = getOption("digits") - 2,
    tadj = 0.65, stats = TRUE, use.n = FALSE, bars = TRUE,
    legend = FALSE, xadj = 1, yadj = 1, bord = FALSE, big.pts = FALSE, ...)

### Arguments

x
- fitted model object of class rpart. This is assumed to be the result of some function that produces an object with the same named components as that returned by the rpart function.

splits
- logical flag. If TRUE (default), then the splits in the tree are labeled with the criterion for the split.

which
- labels splits 1 = center, 2 = left, 3 = right, 4 = both.

label
- a column name of x$frame; values of this will label the nodes. For the "class" method, label="yval" results in the factor levels being used, "yprob" results in the probability of the winning factor level being used, and 'specific yval level' results in the probability of that factor level.

FUN
- the name of a labeling function, e.g. text.

all.leaves
- Logical. If TRUE, all nodes are labeled, otherwise just terminal nodes.

tadj
- Adjustment of text above (or below) splits.

pretty
- an integer denoting the extent to which factor levels in split labels will be abbreviated. A value of (0) signifies no abbreviation. A NULL, the default, signifies using elements of letters to represent the different factor levels.

digits
- number of significant digits to include in numerical labels.

stats
- If TRUE adds statistics to nodes.

use.n
- If TRUE adds N to labels. (#events level1/ #events level2/etc. for class, n for anova, and #events/n for poisson and exp).

bars
- If TRUE adds barplots for multivariate regression trees.

legend
- If TRUE adds legends for multivariate regression trees.

xadj, yadj
- varies the size of barplots for multivariate regression trees.

bord
- Adds borders (boxes) to barplots for multivariate regression trees.

big.pts
- Adds color coded points to nodes. Useful to track groups to PCA plot (see rpart.pca).

...  
- Graphical parameters may also be supplied as arguments to this function (see par).

### Side Effects

the current plot of a tree dendrogram is labeled.

### See Also

text.plot.rpart, rpart.post.rpart, abbreviate
trclcomp

Tree-Clustering Comparison

Examples

```r
data(car.test.frame)
z.auto <- rpart(Mileage ~ Weight, car.test.frame)
plot(z.auto)
text(z.auto, use.n=TRUE, all=TRUE)
```

data(car.test.frame)
z.auto <- rpart(Mileage ~ Weight, car.test.frame)
plot(z.auto)
text(z.auto, use.n=TRUE, all=TRUE)

Description

This function compares the within-group variation for groups formed by tree partitioning and unconstrained clustering. The results are plotted and returned invisibly.

Usage

```r
trclcomp(x, method = "com")
```

Arguments

- `x` : Rpart object with method "mrt" – see `rpart`
- `method` : The clustering method for the unconstrained clustering – see `hclust`.

Details

The within-group variation for groups formed by tree partitioning and unconstrained clustering are compared for all sizes of the hierarchy of tree partitions.

Value

Returns a list (invisibly) of the within-tree and within-cluster variation for all tree sizes.

Author(s)

Glenn De’ath

References


Examples

```r
data(spider)
fit <- mvpart(data.matrix(spider[,1:12])~herbs+reft+moss+sand+twigs+water,spider)
trclcomp(fit)
```
```r
# tree.depth

## Description

Internal function for Rpart

## Details

tree.depth is to be used only internally.

# xdiss

## Description

The function computes extended dissimilarity indices which are for long gradients have better good rank-order relation with gradient separation and are thus efficient in community ordination with multidimensional scaling.

## Usage

```r
xdiss(data, dcrit = 1, dauto = TRUE, dinf = 0.5, method = "man", use.min = TRUE, eps = 1e-04, replace.neg = TRUE, big = 10000, sumry = TRUE, full = FALSE, sq = FALSE)
```

## Arguments

- `data`: Data matrix
- `dcrit`: Dissimilarities < dcrit are considered to have no species in common and are recalculated.
- `dauto`: Automatically select tuning parameters – recommended.
- `method`: Dissimilarity index
- `use.min`: Minimum dissimilarity of pairs of distances used – recommended.
- `dinf, eps, replace.neg, big`: Internal parameters – leave as is usually.
- `sumry`: Print summary of extended dissimilarities?
- `full`: Return the square dissimilarity matrix.
- `sq`: Square the dissimilarities – useful for distance-based partitionong.

## Details

The function knows the same dissimilarity indices as gdist.

## Value

Returns an object of class distance with attributes "Size" and "ok". "ok" is TRUE if rows are not disconnected (De’ath 1999).
```
Author(s)
Glenn De’ath

References

Examples
```r
data(spider)
spider.dist <- xdiss(spider)
```

---

**xpred.rpart**

*Return Cross-Validated Predictions*

**Description**

Gives the predicted values for an *rpart* fit, under cross validation, for a set of complexity parameter values.

**Usage**

```r
xpred.rpart(fit, xval=10, cp)
```

**Arguments**

- **fit** 
  *a rpart object.*
- **xval** 
  number of cross-validation groups. This may also be an explicit list of integers that define the cross-validation groups.
- **cp** 
  the desired list of complexity values. By default it is taken from the `cptable` component of the fit.

**Details**

Complexity penalties are actually ranges, not values. If the `cp` values found in the table were .36, .28, and .13, for instance, this means that the first row of the table holds for all complexity penalties in the range [.36, 1], the second row for `cp` in the range [.28, .36) and the third row for [.13, .28). By default, the geometric mean of each interval is used for cross validation.

**Value**

a matrix with one row for each observation and one column for each complexity value.

**See Also**

*rpart*
Examples

data(car.test.frame)
fit <- rpart(Mileage ~ Weight, car.test.frame)
xmat <- xpred.rpart(fit)
xerr <- (xmat - car.test.frame$Mileage)^2
apply(xerr, 2, sum)  # cross-validated error estimate

# approx same result as rel. error from printcp(fit)
apply(xerr, 2, sum)/var(car.test.frame$Mileage)
printcp(fit)
Index

*Topic datasets
  car.test.frame, 1
  kyphosis, 6
  solder, 35
  spider, 35

*Topic internal
  descendants, 3
  distfull, 4
  eqscplot, 4
  formatg, 4
  model.frame.rpart, 9
  node.match, 11
  rpart.anova, 22
  rpart.branch, 23
  rpart.class, 23
  rpart.dist, 25
  rpart.exp, 26
  rpart.mrt, 27
  rpart.poisson, 30
  rpartco, 31
  snip.rpart.mouse, 34
  string.bounding.box, 36
  sub.barplot, 36
  tree.depth, 40

*Topic manip
  scaler, 32

*Topic methods
  rpart.object, 28

*Topic multivariate
  cmds.diss, 2
  gdist, 5
  scaler, 32
  trclcomp, 39
  xdist, 40

*Topic tree
  labels.rpart, 7
  meanvar.rpart, 8
  mvpart, 9
  na.rpart, 10
  path.rpart, 11
  plot.rpart, 12
  plotcp, 13
  post.rpart, 14
  predict.rpart, 16
  print.rpart, 17
  printcp, 18
  prune.rpart, 19
  residuals.rpart, 20
  rpart, 21
  rpart.control, 24
  rpart.matrix, 27
  rpart.object, 28
  rpart.pca, 29
  rpconvert, 31
  rsq.rpart, 32
  snip.rpart, 33
  summary.rpart, 37
  text.rpart, 37
  xpred.rpart, 41

abbreviate, 8, 15, 38

car.test.frame, 1
cmds.diss, 2
cmdscale, 2
descendants, 3
dist, 5, 6
distfull, 4
eqscplot, 4
formatg, 4
gdist, 2, 3, 5, 21, 40
hclust, 39
kyphosis, 6
labels.rpart, 7, 17
meanvar(meanvar.rpart), 8
meanvar.rpart, 8
model.frame.rpart, 9
mvpart, 2, 9
na.rpart, 10
node.match, 11
INDEX

path.rpart, 11
plot.rpart, 8, 12, 15, 34, 38
plotcp, 13
post (post.rpart), 14
post.rpart, 14, 38
pred.rpart (predict.rpart), 16
predict, 17
predict.rpart, 16
print, 18
print.rpart, 17, 22
printcp, 14, 18, 18, 37
prune (prune.rpart), 19
prune.rpart, 19
residuals.rpart, 20
rpart, 2, 3, 10, 12–15, 19, 21, 24, 25, 29, 30,
38, 39, 41
rpartanova, 22
rpart.branch, 23
rpart.class, 23
rpart.control, 22, 24
rpart.dist, 25
rpart.exp, 26
rpart.matrix, 27
rpart.mrt, 27
rpart.object, 14, 17, 18, 22, 28, 37
rpart.pca, 10, 29
rpart.poisson, 30
rpartcallback (rpart), 21
rpartco, 31
rpconvert, 31
rsq.rpart, 32
scaler, 32
snip.rpart, 33
snip.rpart.mouse, 34
solder, 35
spider, 35
string.bounding.box, 36
sub.barplot, 36
summary, 37
summary.rpart, 18, 22–24, 26, 27, 30, 37
text, 38
text.rpart, 13, 15, 23, 24, 26, 27, 30, 37
trclcomp, 39
tree, 16
tree.depth, 40
xdiss, 2, 21, 40
xpdr.rpart, 41