Package 'treeClust'

May 12, 2018

Version 1.1-7

Date 2018-05-07

Title Cluster Distances Through Trees

Author Sam Buttrey

Maintainer Sam Buttrey <buttrey@nps.edu>

Depends rpart, cluster

Suggests parallel

Description Create a measure of inter-point dissimilarity useful
for clustering mixed data, and, optionally, perform the clustering.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2018-05-12 03:24:41 UTC

R topics documented:

cramer *Compute Cramer's V for a two-way table*

Description

This function computes the value of Cramer's V for a two-way table.

Usage

```
cramer(tbl)
```

Arguments

tbl	Two-way table, or matrix, of counts.

Details

If X^2 is the usual chi-squared measure of association in a two-way table, Cramer's V is sqrt (X^2 / (n * (k-1))), where n is the total number of observations in the table, and k is min (nrow(table), ncol(table)).

Value

Numeric value of Cramer's V, with name "X-squared".

Author(s)

Sam Buttrey

References

Agresti, "Categorical Data Analysis," p. 75, where V^2 is used.

d3.dist *D3-style dissimilarity for a single tree*

Description

Compute the set of pairwise dissimilarities across all observations in a tree. Each dissimilarity measures the extent to which observations are "far apart" in the tree: the dissimilarity is 0 if the pair land in the same leaf, 1 if they land on leaves that have only the root as common ancestors, and otherwise something intermediate.

Usage

```
d3.dist(mytree, return.pd = FALSE)
```

Arguments

mytree	Output from "tree"
return.pd	If TRUE return the matrix of pairwise distances among leaves. Useful for debugging. Default FALSE.

Details

Two observations have distance 0 if they fall in the same leaf; otherwise, the distance measures the ratio of the deviance of a tree trimmed so that they do fall in the same leaf to the deviance of the original tree.

Value

Item of class "dist" giving inter-point distances.

Author(s)

Sam Buttrey

See Also

treeClust

---

leaf.numbers *Convert "where" entry of tree frame into leaf numbers*

Description

The "where" entry of a tree object denotes leaves by row numbers in the "frame" object. This converts those to actual leaf numbers.

Usage

```
leaf.numbers(tree)
```

Arguments

tree	Item of class "tree".

Value

Vector, the same length as tree$where, giving leaf numbers.

Author(s)

Sam Buttrey

make.leaf.paths *Make matrix of leaf paths*

Description

It is helpful to know the parent nodes for each tree node. This function creates a matrix with that information.

Usage

```
make.leaf.paths(up.to = 2047)
```

Arguments

up.to	Number of rows for which to compute leaf.paths.

Details

The ith row of the resulting matrix lists all the leaves, including i, that would be traversed from the root to leaf i. Unneeded columns have zeros.

Value

Numeric matrix with "up.to" rows. If 2^j <= up.to < 2^(j+1), j columns.

plot.treeClust *Plot treeClust object*

Description

Plot a picture of a treeClust object. This picture shows the deviance ratio on the vertical axis, scaled to have maximum 1, and the tree index on the horizontal. Each point is shown by a digit (or digits) giving the size of the tree.

Usage

```
## S3 method for class 'treeClust'
 plot(x, extended, ...)
```

Arguments

x	Object of class treeClust
extended	Logical. If TRUE, include all variables, even those whose trees were dropped. Otherwise only include variables whose trees were kept. Default TRUE.
...	Other arguments to be passed to the plot function.

Value

None. The side effect is that the plot is produced on the current device.

`print.treeClust` *Print treeClust object*

Description

Print some details about a treeClust object, and the "tbl" element.

Usage

```
## S3 method for class 'treeClust'
 print(x, ...)
```

Arguments

`x`	Object of class treeClust
`...`	Other stuff.

Value

None. The "tbl" element is printed to the screen.

`rp.deviance` *Compute deviance within nodes of classification trees*

Description

An rpart regression tree carries the deviance around (in the frame$dev element). This function computes the deviance for classification trees.

Usage

```
rp.deviance(x, ...)
```

Arguments

`x`	An object of class rpart
`...`	Other arguments, currently unused.

Details

For a vector of leaf counts n whose sum is N, the deviance is (-2) times the sum of n log (n/N), taking 0 log 0 as 0.

Value

Vector of deviances for every row in the tree's frame.

Author(s)

Sam Buttrey

See Also

rpart

`rpart.predict.leaves` *Return the leaf into which observations are predicted to fall*

Description

The "where" element of an rpart object gives the leaf into which each observation used building the tree falls. This produces the equivalent for new data.

Usage

```
rpart.predict.leaves(rp, newdata, type = "where")
```

Arguments

`rp`	Object of class rpart.
`newdata`	New data frame, with the columns used in the rpart model.
`type`	Style of leaf identification: "where" or "leaf"

Details

There are two ways to identify the leaf into which an observation falls. The way used in the "where" element of an rpart object is to give the row number of the leaf within the object's "frame" element. That is the approach used here when type = "where". When type = "leaf" the actual leaf number is returned. For example, in a tree where node 2 is a terminal node and node 3 splits into terminal nodes 6 and 7, type = "leaf" will return a vector with values 2, 6 and 7. Type = "where" will return a vector with values 2, 4 and 5, since rows 2, 4 and 5 of the tree's "frame" element are leaves.

Value

If type = "where", numeric vector of row numbers describing leaves in the tree's "frame" component. If type = "leaf," character vector of leaf numbers.

Author(s)

Sam Buttrey

See Also

rpart

`summary.treeClust` *Summarize treeClust object*

Description

Print some details about a treeClust object.

Usage

```
## S3 method for class 'treeClust'
 summary(object, ...)
```

Arguments

`object`	Object of class treeClust
`...`	Other stuff.

Value

None. A few lines of information are printed to the screen.

`tcdist` *Compute treeClust dissimilarities*

Description

Given a treeClust object, or the necessary components, compute all pairwise dissimilarities for input to a clustering algorithm

Usage

```
tcdist(obj, d.num = 1, tbl, mat, trees, verbose=0)
```

Arguments

`obj`	Object of class treeClust
`d.num`	Method of dissimilarities computation. See "Details".
`tbl`	Two-column of information about trees. Always included in a treeClust object, but may be supplied separately. Required if d.num = 2 or 4.
`mat`	Matrix of leaf-membership factors, if not supplied in "obj".
`trees`	List of trees, if not supplied in obj.
`verbose`	If > 0, print some information useful for debugging.

Details

There are four ways to compute inter-point dissimilarities from a treeClust object. If d.num = 1, two points differ by the number of trees in which they land in different leaves. "Mat" is required. If d.num = 2, the computation for d.num = 1 is used, but each tree gets a different weight. "Mat" and "tbl" are required.tbl" are required.

The computation for d.num = 3 requires that the set of trees be supplied. With this approach two observations differ, on a particular tree, according to how far apart they are on that tree. For d.num = 4, both tree and "tbl" are required; this is a weighted version of the d.num = 3 dissimilarity.

Value

Object of class "dist" giving pairwise distances for the original data used to build the treeClust object.

Author(s)

Sam Buttrey

See Also

`treeClust`

---

tcnewdata *Create all-numeric data to mimic the inter-point distances from treeClust*

Description

treeClust produces a vector of dissimilarities, but these objects are large. This function produces a data frame of data whose inter-point distances are related to the treeClust ones, for use in, for example, k-means.

Usage

```
tcnewdata(obj, d.num = 1, tbl, mat, trees)
```

Arguments

`obj`	Output from a call to `treeClust`.
`d.num`	Integer, 1-4, describing dissimilarity algorithm. See `treeClust`.
`tbl`	Matrix of tree deviances and sizes, if not present in `obj`.
`mat`	Matrix of leaf memberships, if not present in `obj`.
`trees`	List of trees, if not present in `obj` (needed for d.num = 3 or 4),

Details

See the paper by Buttrey and Whitaker. The inter-point distances of this data set "mirror" the treeClust distances, but only if they are computed in a particular non-standard way. This is experimental.

Value

Numeric matrix of data whose inter-point distances match the d1 distances computed by treeClust, and which may be useful for d2-d4 as well.

Author(s)

Sam Buttrey, buttrey@nps.edu

References

Buttrey and Whitaker, The R Journal, 7/2, 2015.

See Also

treeClust

`treeClust` *Build a tree-based dissimilarity for clustering, and optionally perform the clustering*

Description

This function uses a set of classification or regression trees to build an inter-point dissimilarity in which two points are similar when they tend to fall in the same leaves of trees. The user can pass in a clustering algorithm and/or ask for the dissimilarities or the set of trees.

Usage

```
treeClust(dfx, d.num = 1, col.range = 1:ncol(dfx), verbose = F,
  final.algorithm, k, control = treeClust.control(), rcontrol = rpart.control(), ...)
```

Arguments

`dfx`	Input data frame. Columns may be numeric or categorical. Missing values are permitted.
`d.num`	Integer: Dissimilarity specifier. When d.num = 1, the dissimilarity between two observations is the proportion of trees where they disagree. With d.num = 2, those counts are weighted according to tree quality. In d.num = 3, dissimilarities are variable with trees, reflecting the belief that some pairs of leaves are closer together than others. With d.num = 4, those dissimilarities are weighted by tree quality.

col.range	Integer: the indices of the columns used. Defaults to all.
verbose	If non-zero, print degugging messages to the screen.
final.algorithm	Final algorithm, to be used to cluster the computed distances. This may be "pam", "agnes", "clara" or "kmeans".
k	If final.algorithm is supplied, the number of clusters is required.
control	List of the sort produced by treeClust.control, giving specifications for the fitting routine.
rcontrol	List of the sort produced by rpart.control, giving arguments for the rpart routine.
...	Other arguments, to be passed to the final clustering algorithm if specified.

Details

The treeClust approach builds a set of classification or regresion trees, one for each variable. Trees are pruned, and those that are pruned to the root are discarded. For each remaining tree, an observation's leaf membership serves as the starting point for a dissimilarity measurement.

Value

If control$cluster.only is TRUE, a vector of cluster assignments, as produced by the final algorthm. Otherwise, a list with these items:

call	The call that produced the object
d.num	d.num, as supplied
tbl	Two-column matrix with one row for each tree retained, giving size and deviance ratio
extended.tbl	Two-column matrix like tbl, but with one row for every variable, giving size and deviance ratio (these will be 1 and 0 for variables whose trees were discarded
final.algorithm	final.algorithm, as supplied
final.clust	If final.algorithm is supplied, the output from the final clustering algorithm; otherwise, NULL
additional.args	Any additional arguments specified
tree	If control$return.trees is TRUE, a list holding all the retained trees. This can make the resulting object very large.
dists	If control$return.dists is TRUE, an object of class dist with the set of pairwise inter-point dissimilarities
mat	If control$return.mat is TRUE, a data frame. If final.algorithm is "pam" or "agnes" this contains leaf assignment indices. Otherwise this holds a dataset useful as input to k-means or clara. Experimental.

Author(s)

Sam Buttrey, buttrey@nps.edu

References

Buttrey and Whitaker, "treeClust: An R Package for Tree-Based Clustering Dissimilarities," The R Journal, 7/2, 2015.

See Also

treeClust.control

Examples

```
iris.km6 <- treeClust (iris[,-5], d.num = 2, final.algorithm = "kmeans", k=6)
table (iris.km6$final.clust$cluster, iris$Species)
```

---

treeClust.control *Parameters describing the output from a treeClust fit*

Description

This function produces a list that is used as input to treeClust to determine which items are preserved in the output.

Usage

```
treeClust.control(return.trees = FALSE, return.mat = TRUE,
 return.dists = FALSE, return.newdata = FALSE, cluster.only = FALSE,
 serule = 0, DevRatThreshold = 1, parallelnodes = 1, ...)
```

Arguments

return.trees	If TRUE, all the trees that go into the object are returned. This can make the treeClust object very large. Default FALSE.
return.mat	If TRUE, return a matrix describing leaf membership. Default TRUE.
return.dists	If TRUE, return an object of class 'dissimilarity' giving all pairwise distances between observations. This can be very large for large datasets. Default FALSE.
cluster.only	If TRUE, return only the clustering vector, which names the cluster into which each observation is places. Default FALSE.
return.newdata	If TRUE, return a numeric matrix describing leaf membership and/or inter-point distance (see "Details"). Default FALSE.
serule	Describes how to prune the rpart trees. By default, each tree is pruned to the minimum error size. With serule > 0, each tree is pruned to the smallest size for which the cross-validated error is less than (min error) + (serule * sds).
DevRatThreshold	Trees whose deviance ratio is greater than this number are presumed to have arisen from redundant variables. The predictor at the tree's root is dropped, a new tree built, and the new deviance ratio computed. this process is repeated until the resulting tree has deviance ratio less than or equal to the threshold. Default: 1 (do not drop any such trees).

parallelnodes	Describes whether to use parallel processing by creating a "computing cluster" containing "parallelnodes" nodes. If that number is = 1 no cluster is created. Here "cluster" is referring to a set of nodes operating in parallel, not to the clustering of the data.
...	Other arguments, passed onto the output.

Details

The "newdata" item is a numeric matrix that gives inter-point distances whose form depends on the "d.num" argument to treeClust(). When d.num = 1, each tree contributes a set of 0-1 dummy variables that serve as leaf membership indicators, and with d.num = 2, each tree's indicators are multiplied by that tree's "strength." With d.num = 3, a tree with k leaves contributes k-choose-2 columns, with the distances between distinct rows matching the d3 distances, and likewise with d.num = 4, a tree with k leaves produced k-choose-2 columns that have been weighted by tree strength.

Value

list, with all the input arguments and their supplied or default values.

Author(s)

Sam Buttrey, buttrey@nps.edu

See Also

treeClust

---

treeClust.dist *Built treeClust distance*

Description

This function uses treeClust to build a distance. It is intended to act analagously to daisy and dist.

Usage

```
treeClust.dist(x, ...)
```

Arguments

x	Data set from which to compute distances via treeClust.
...	Other argments to be passed to treeClust.

Details

The `treeClust` function's first argument is named dfx. This calls the same code, but by naming the first argument `x` it allows users to employ this function interchangeably with `dist` and `daisy`, which expect arguments named `x`. This function also sets the `return.dists` flag and extract the distance object so that that is the only thing returned.

Value

An object of class `dissimilarity`.

Author(s)

Sam Buttrey

See Also

`treeClust`

`treeClust.rpart` *Build an rpart tree as part of treeClust*

Description

This function builds one tree, as part of a treeClust analysis. It will not normally be called by users.

Usage

```
treeClust.rpart(i, dfx, d.num, control, rcontrol)
```

Arguments

`i`	Index of column number (in dfx) of response variable.
`dfx`	Data set used to build tree
`d.num`	Distance number, 1-4, describing measurement for clustering.
`control`	List of controls for treeClust, often output of treeClust.control().
`rcontrol`	List of controls for rpart, often output of rpart.control().

Details

It is useful to encapsulate some of the tree-building code so that it can be used either in a loop or in parallel.

Value

List containing some of these elements (below). Size and DevRatio are always present.

`DevRat`	Deviance ratio (decrease in dev. / original dev.) for this tree; always present
`Size`	Size of pruned tree. If no tree is grown, Size is 1.
`tree`	The pruned tree, if needed
`leaf.where`	Vector of leaf membership indices, if Size > 1

Author(s)

Sam Buttrey

See Also

`treeClust`

Index