Package ‘cccd’

February 19, 2015

Version 1.04
Date 2013-10-29
Title Class Cover Catch Digraphs
Author David J. Marchette <dmarchette@gmail.com>
Maintainer David J. Marchette <dmarchette@gmail.com>
Depends igraph, plotrix, proxy
Suggests deldir, Matrix
Description Class Cover Catch Digraphs, neighborhood graphs, and relatives.
License GPL (>= 2)
Repository CRAN
Date/Publication 2013-10-29 16:58:01
NeedsCompilation no

R topics documented:

cccd ................................................................. 2
cdd ................................................................. 4
dominate ......................................................... 5
gg ................................................................. 6
juggling ......................................................... 8
nng ............................................................... 9
prune ............................................................ 10
rng ............................................................... 11

Index 13
Description

Constructs a class cover catch digraph from points or interpoint distance matrices.

Usage

cccd(x = NULL, y = NULL, dxx = NULL, dyx = NULL, method = NULL)
cccd.rw(x=NULL,y=NULL,dxx=FALSE,dyx=NULL,method=NULL,m=1,d=2)
cccd.classifier(x,y,method=NULL)
cccd.classify(data, C,method=NULL)
cccd.classifier.rw(x,y,m=1,d=2)
cccd.multiclass.classifier(data, classes)
cccd.multiclass.classify(data,C,method=NULL)

## S3 method for class 'cccd'

plot(x, ..., plot.circles = FALSE, dominate.only = FALSE,
     D = NULL, vertex.size = 2, vertex.label = NA,
     vertex.color = "SkyBlue2", dom.color = "Blue",
     ypch = 20, ycex = 1.5, ycol = 2,
     use.circle = FALSE, balls = FALSE,
     ball.color = gray(0.8), square = FALSE, xlim, ylim)

Arguments

x, y the target class and non-target class points. Either x, y or dxx, dyx must be provided. In the case of plot, x is an object of class cccd.
dxx, dyx interpoint distances (x against x and y against x). If these are not provided they are computed using x and y.
method the method used for the distance. See dist.
m slope of the null hypothesis curve
data data to be classified
dimension of the data
classes class labels of the data
C cccd object
plot.circles logical. Plot the circles around the points if TRUE.
dominate.only logical. Only plot the digraph induced by the dominating set.
D a dominating set. Only used if dominate.only is TRUE. If dominate.only is TRUE and D is NULL, then dominate is called.
vertex.size, vertex.color, vertex.label, dom.color parameters controlling the plotting of the vertices. dom.color is the color of the vertices in the dominating set.
balls, ball.color
    if balls=TRUE, the cover is plotted as filled balls, with ball.color controlling
    their color.
ypch, ycex, ycol  parameters for plotting the non-target points.
use.circle.radii
    logical. Ensure that the circles fit within the plot.
square
    logical. Make the plot square.
xlim, ylim
    if present, these control the plotting region.
...  arguments passed to plot or plot.cccd.

Details

The class cover catch digraph is a graph with vertices defined by the points of x and edges defined according to the balls \( B(x, d(x, Y)) \). There is an edge between vertices \( x_1, x_2 \) if \( x_2 \in B(x_1, d(x_1, Y)) \).

Value

an object of class igraph. In addition, it contains the attributes:

\( R \)
    a vector of radii.
\( Y \)
    the y vectors.
layout
    the x vectors.

Note

The plotting assumes the cccd used Euclidean distance, and so the balls/circles will be Euclidean balls/circles. If the method used in the distance was some other metric, you’ll have to plot the balls/circles yourself if you want them to be correct on the plot.

Author(s)

David J. Marchette, david.marchette@navy.mil

References


See Also

ccd, rng, gg, dist, plot.cccd
Examples

```r
set.seed(456330)
z <- matrix(runif(100),nrow=2)
ind <- which(z[,1]<.5 & z[,2]<.5)
x <- z[ind,]
y <- z[-ind,]
C <- cccd(x,y)
z2 <- matrix(runif(100),nrow=2)
ind <- which(z2[,1]<.5 & z2[,2]<.5)
cls <- rep(0,nrow(z2))
cls[ind] <- 1
out <- cccd.classify(z2,C)
sum(out != cls)/nrow(z2)
## Not run:
plot(g,plot.circles=TRUE,dominate.only=TRUE)
points(z2,col=2*(1-cls)+1,pch=20)
## End(Not run)
```

---

**ccd**

*Cluster Catch Digraphs*

**Description**

construct the cluster catch digraph from a data matrix.

**Usage**

```r
ccd(data, m = 1, alpha = 0.05, sequential = TRUE, method = NULL)
```

## S3 method for class 'ccd'

```r
plot(x,...)
```

**Arguments**

- `data`: a matrix of observations.
- `m`: slope of the null hypothesis curve.
- `alpha`: alpha for the K-S test if sequential=T.
- `sequential`: use the sequential or non-sequential version.
- `method`: the method used for the distance. See `dist`.
- `x`: an object of class `ccd`.
- `...`: arguments passed to `plot.ccd`.

**Details**

cluster cover digraph. `plot.ccd` is just a call to `plot.cccd`. 
Value

an object of class igraph. In addition, this contains the attributes:

- `R`     the radii.
- `stats` the K-S statistics.
- `layout` the data vectors.
- `walks` the y-values of the random walks.
- `fs`    the null hypothesis curve.
- `A`     the adjacency matrix.
- `m, a`  arguments passed to ccd.

Author(s)

David J. Marchette david.marchette@navy.mil

References


See Also

cccd

Examples

```r
x <- matrix(rnorm(100), ncol=2)
G <- ccd(x)
## Not run:
plot(G)
## End(Not run)
```

---

**dominate**

*Dominating Sets*

Description

find maximum dominating sets in (di)graphs.

Usage

dominate(g, method = "greedy")
Arguments

- `g`: an adjacency matrix.
- `method`: one of "greedy", "random", "byRadius".

Details

dominate is the main program which calls the others, as indicated by method. Greedy is the greedy dominating algorithm. In the greedy method ties are broken by first index (a la which.max). The byRadius method uses the radii to break ties while the random routine breaks ties randomly.

Value

a vector of vertices corresponding to the dominating set. Note: just like the vertex labels in igraph, these are 0-based.

Author(s)

David J. Marchette david.marchette@navy.mil

References


Examples

```r
x <- matrix(runif(100),ncol=2)
y <- matrix(runif(100,-2,2),ncol=2)
G <- cccd(x,y)
D <- dominate(G)
## Not run:
plot(G,balls=TRUE,D=D)
## End(Not run)
```

---

### Gabriel Graph

Description

A Gabriel graph is one where the vertices are points and there is an edge between two points if the maximal ball between the points contains no other points.

Usage

```r
gg(x, r = 1, method = NULL, usedeldir = TRUE)
```
Arguments

- **x**: a matrix of observations.
- **r**: a multiplier on the ball radius.
- **method**: the method used for the distance. See `dist`
- **usedeldir**: logical. Whether to use the `deldir` package or not.

Details

places an edge between two points \(i, j\) if the ball centered between the points with radius \(rd(i, j)/2\) contains no other points.

Value

an object of class igraph. In addition it contains the attributes:

- **layout**: the data.
- **r, p**: arguments passed to `gg`

Author(s)

David J. Marchette

References


See Also

`rng`, `dist`

Examples

```r
x <- matrix(runif(100), ncol=2)

g <- gg(x)
## Not run:
plot(g)
## End(Not run)
```
juggling  Juggling

Description

a resampled version of the CCCD classifier.

Usage

juggle(data, classes, sampled = TRUE, sample.dim = FALSE,
       num = 100, sample.proportion = 0.1, k = 2, method = NULL)

juggle.classify(data,J,tdata,indices)

Arguments

- **data, tdata**: training data from which to build the classifier. In the case of `juggle.classify`, `tdata` is the training data and `data` is the test data.
- **classes**: class labels.
- **sampled**: whether the data are subsampled.
- **sample.dim**: if TRUE, the dimensions (variates) are also sampled.
- **num**: number of juggles (resamples).
- **sample.proportion**: proportion of the data to sample. If 1 or greater, the data are sampled with replacement.
- **k**: number of variates to sample when `sample.dim` is TRUE.
- **J**: the juggled classifier.
- **indices**: the indices of the juggles to use.
- **method**: the method used for the distance. See `dist`.

Details

The idea of juggling is to sample the data, compute a CCCD classifier, then repeat. The resampling is controlled by the two sampling variables, which basically determine whether the data are sampled with replacement, or whether a subsample is used. If `sample.dim` is TRUE, the variates are also sampled, with `k` indicating how many are sampled.

Value

`juggle.classify` returns a matrix holding the classification probabilities for each observation in `data`. a list consisting of:

- **S**: the dominating sets.
- **R**: the radii.
- **dimension**: the dimension of the data.
in the case of `sample_dim=TRUE`, the variables sampled each time.

Only the indicies into the training data are stored in `j`, which is why the classifier requires the original training data in `tdata`.

**Author(s)**

David J. Marchette, david.marchette@navy.mil

**See Also**

`cccd`, `dist`

---

**nng**

*Nearest Neighbor Graphs*

**Description**

nearest neighbor, k-nearest neighbor, and mutual k-nearest neighbor (di)graphs.

**Usage**

```r
nng(x = NULL, dx = NULL, k = 1, mutual = FALSE, method = NULL)
```

**Arguments**

- `x` a data matrix. Either `x` or `dx` is required
- `dx` interpoint distance matrix
- `k` number of neighbors
- `mutual` logical. if true the neighbors must be mutual. See details.
- `method` the method used for the distance. See `dist`

**Details**

A k-nearest neighbor graph is a digraph where each vertex is associated with an observation and there is a directed edge between the vertex and it's k nearest neighbors. A mutual k-nearest neighbor graph is a graph where there is an edge between x and y if x is one of the k nearest neighbors of y AND y is one of the k nearest neighbors of x.

**Value**

an object of class `igraph` with the extra attributes

- `layout` the x vectors.
- `k, mutual, p` arguments given to `nn`.
**prune**

**Prune Points**

a nearest neighbor pruning using neighborhood graphs.
Usage
prune(x, classes, prox = "Gabriel", ignore.ties = TRUE, ...)

Arguments
x a data matrix.
classes a vector of class labels.
prox type of proximity graph.
ignore.ties do not prune if there is a tie vote.
... arguments passed to the proximity graph.

Details
First a proximity graph is computed on the data. Then points are marked if their neighbors have a different class than they do: if the most common class among the neighbors is different than the point. Then all marked points are removed.

Value
A list with attributes:
x the pruned data.
v the indices of the retained data.
g the proximity graph.

Author(s)
David J. Marchette, david.marchette@navy.mil

References

rng

Relative Neighborhood Graph.

Description
the relative neighborhood graph defined by a set of points.

Usage
rng(x=NULL, dx=NULL, r = 1, method = NULL, usedeldir=TRUE)
Arguments

x a data matrix. Either x or dx must be provided.
dx an interpoint distance matrix.
r a multiplier to grow the balls.
method the method used for the distance. See dist
usedeldir a logical. If true and the data are two dimensional and the deldir package is
installed, the Delaunay triangularization is first computed, and this is used to
compute the relative neighborhood graph.

Details

the relative neighborhood graph is defined in terms of balls centered at observations. For two
observations, the balls are set to have radius equal to the distance between the observations (or
r times this distance if r is not 1). There is an edge between the vertices associated with the
observations if and only if there are no vertices in the lune defined by the intersection of the balls.

Value

an object of class igraph, with the additional attributes
layout the x matrix.
r, p arguments given to rng.

Author(s)

David J. Marchette david.marchette@navy.mil

References

J.W. Jaromczyk and G.T. Toussaint, "Relative neighborhood graphs and their relatives", Proceed-
ings of the IEEE, 80, 1502-1517, 1992.

See Also

gg, cccd, ccd, dist

Examples

x <- matrix(runif(100), ncol=2)
g <- rng(x)
## Not run:
plot(g)

## End(Not run)
Index

*Topic graphs
  cccd, 2
  ccd, 4
  dominate, 5
  gg, 6
  juggling, 8
  nng, 9
  prune, 10
  rng, 11
  cccd, 2, 5, 9, 12
  ccd, 3, 4, 12
  dist, 2–4, 7–10, 12
  dominate, 5
  gg, 3, 6, 12
  juggle (juggling), 8
  juggling, 8
  nng, 9
  plot.cccd, 3
  plot.cccd (cccd), 2
  plot.cccd (ccd), 4
  prune, 10
  rng, 3, 7, 11