Package ‘sparr’

March 10, 2016

Type Package

Title SPAtial Relative Risk

Version 0.3-8

Date 2016-03-11

Author T.M. Davies, M.L. Hazelton and J.C. Marshall

Maintainer Tilman M. Davies <tdavies@maths.otago.ac.nz>

Description Provides functions to estimate kernel-
smoothed relative risk functions and perform subsequent inference.

Depends R (>= 2.10.1), spatstat

Imports rgl, MASS

License GPL (>= 2)

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2016-03-10 23:32:16

R topics documented:

sparr-package 2
as.im.bivden 5
bivariate.density 6
KBivN 11
KBivQ 12
LSCV.density 13
LSCV.risk 15
NS 17
OS 19
PBC 20
plot.bivden 21
risk 23
summary.bivden 26
summary.rrs 27
tolerance 27

Index 32

sparr-package *The sparr Package: SPAtial Relative Risk*

Description

Provides functions to estimate fixed and adaptive kernel-smoothed relative risk surfaces via the density-ratio method and perform subsequent inference.

Details

Package:	sparr
Version:	0.3-8
Date:	2016-03-11
License:	GPL (>= 2)

Kernel smoothing, and the flexibility afforded by this methodology, provides an attractive approach to estimating complex probability density functions. This is particularly of interest when exploring problems in geographical epidemiology, the study of disease dispersion throughout some spatial region, given a population. The so-called 'relative risk surface', constructed as a ratio of estimated case to control densities (Bithell, 1990; 1991), describes the variation in the 'risk' of the disease, given the underlying at-risk population. This is a technique that has been applied successfully for mainly exploratory purposes in a number of different examples (see for example Sabel et al., 2000; Prince et al., 2001; Wheeler, 2007).

This package provides functions for bivariate kernel density estimation (KDE), implementing both fixed and 'variable' or 'adaptive' (Abramson, 1982) smoothing parameter options (see the function documentation for more information). A selection of bandwidth calculators for bivariate KDE and the relative risk function are provided, including one based on the maximal smoothing principle (Terrell, 1990), and others involving a leave-one-out least-squares cross-validation (see below). In addition, the ability to construct asymptotically derived p-value surfaces ('tolerance' contours of which signal statistically significant sub-regions of 'extremity' in a risk surface - Hazelton and Davies, 2009; Davies and Hazelton, 2010), as well as some flexible visualisation tools, are provided.

The content of `sparr` can be broken up as follows:

Datasets
`PBC` a case/control planar point pattern (`ppp`) concerning liver disease in northern England. Also available is the case/control dataset `chorley` of the `spatstat` package, which concerns the distribution of laryngeal cancer in an area of Lancashire, England.

Bandwidth calculators
`OS` estimation of an isotropic smoothing parameter for bivariate KDE, based on the oversmoothing principle introduced by Terrell (1990).
`NS` estimation of an isotropic smoothing parameter for bivariate KDE, based on the optimal value for a normal density (bivariate normal scale rule - see e.g. Wand and Jones, 1995).

LSCV.density a least-squares cross-validated (LSCV) estimate of an isotropic bandwidth for bivariate KDE (see e.g. Bowman and Azzalini, 1997).
LSCV.risk a least-squares cross-validated (LSCV) estimate of a jointly optimal, common isotropic case-control bandwidth for the kernel-smoothed risk function (see Kelsall and Diggle, 1995a;b and Hazelton, 2008).

Bivariate functions
KBivN bivariate normal (Gaussian) kernel
KBivQ bivariate quartic (biweight) kernel
bivariate.density kernel density estimate of bivariate data; fixed or adaptive smoothing

Relative risk and p-value surfaces
risk estimation of a (log) relative risk function
tolerance calculation of asymptotic p-value surface

Printing and summarising objects
S3 methods (print.bivden, print.rrs, summary.bivden and summary.rrs) are available for the bivariate density and risk function objects.

Visualisation
Most applications of the relative risk function in practice require plotting the relative risk within the study region (especially for an inspection of tolerance contours). To this end, sparr provides a number of different ways to achieve attractive and flexible visualisation. The user may produce a heat plot, a perspective plot, a contour plot, or an interactive 3D perspective plot (that the user can pan around and zoom - courtesy of the powerful rgl package; see below) for either an estimated relative risk function or a bivariate density estimate. These capabilities are available through S3 support of the plot function; see
plot.bivden for visualising a single bivariate density estimate from bivariate.density, and
plot.rrs for visualisation of an estimated relative risk function from risk.

Dependencies

The sparr package depends upon/imports some other important contributions to CRAN in order to operate; their uses here are indicated:

spatstat - Fast-fourier transform assistance with fixed and adaptive density estimation, as well as region handling; see Baddeley and Turner (2005).
rgl - Interactive 3D plotting of densities and surfaces; see Adler and Murdoch (2009).
MASS - Utility support for internal functions; see Venables and Ripley (2002).

Citation

To cite use of sparr in publications, the user may refer to the following work:
Davies, T.M., Hazelton, M.L. and Marshall, J.C. (2011), sparr: Analyzing spatial relative risk using fixed and adaptive kernel density estimation in R, *Journal of Statistical Software* **39**(1), 1-14.

Author(s)

T.M. Davies
Dept. of Mathematics & Statistics, University of Otago, Dunedin, New Zealand;
M.L. Hazelton and J.C. Marshall
Institute of Fundamental Sciences - Statistics, Massey University, Palmerston North, New Zealand.

Maintainer: T.M.D. `<tdavies@maths.otago.ac.nz>`
Feedback welcomed.

References

Abramson, I. (1982), On bandwidth variation in kernel estimates — a square root law, *Annals of Statistics*, **10**(4), 1217-1223.
Adler, D. and Murdoch, D. (2009), rgl: 3D visualization device system (OpenGL). R package version 0.87; URL: http://CRAN.R-project.org/package=rgl
Baddeley, A. and Turner, R. (2005), Spatstat: an R package for analyzing spatial point patterns, *Journal of Statistical Software*, **12**(6), 1-42.
Bithell, J.F. (1990), An application of density estimation to geographical epidemiology, *Statistics in Medicine*, **9**, 691-701.
Bithell, J.F. (1991), Estimation of relative risk function,. *Statistics in Medicine*, **10**, 1745-1751.
Bowman, A.W. and Azzalini, A. (1997), *Applied Smoothing Techniques for Data Analysis: The Kernel Approach with S-Plus Illustrations.* Oxford University Press Inc., New York. ISBN 0-19-852396-3.

Davies, T.M. and Hazelton, M.L. (2010), Adaptive kernel estimation of spatial relative risk, *Statistics in Medicine*, **29**(23) 2423-2437.
Davies, T.M., Jones, K. and Hazelton, M.L. (2015), Symmetric adaptive smoothing regimens for estimation of the spatial relative risk function, *Submitted for publication*.
Hazelton, M. L. (2008), Letter to the editor: Kernel estimation of risk surfaces without the need for edge correction, *Statistics in Medicine*, **27**, 2269-2272.
Hazelton, M.L. and Davies, T.M. (2009), Inference based on kernel estimates of the relative risk function in geographical epidemiology, *Biometrical Journal*, **51**(1), 98-109.
Kelsall, J.E. and Diggle, P.J. (1995a), Kernel estimation of relative risk, *Bernoulli*, **1**, 3-16.
Kelsall, J.E. and Diggle, P.J. (1995b), Non-parametric estimation of spatial variation in relative risk, *Statistics in Medicine*, **14**, 2335-2342.
Prince, M. I., Chetwynd, A., Diggle, P. J., Jarner, M., Metcalf, J. V. and James, O. F. W. (2001), The geographical distribution of primary biliary cirrhosis in a well-defined cohort, *Hepatology* **34**, 1083-1088.
Sabel, C. E., Gatrell, A. C., Loytonenc, M., Maasiltad, P. and Jokelainene, M. (2000), Modelling exposure opportunitites: estimating relative risk for motor disease in Finland, *Social Science & Medicine* **50**, 1121-1137.
Terrell, G.R. (1990), The maximal smoothing principle in density estimation, *Journal of the American Statistical Association*, **85**, 470-477.
Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S*, Fourth Edition, Springer, New York.
Wand, M.P. and Jones, C.M., 1995. *Kernel Smoothing*, Chapman & Hall, London.
Wheeler, D. C. (2007), A comparison of spatial clustering and cluster detection techniques for

childhood leukemia incidence in Ohio, 1996-2003, *International Journal of Health Geographics*, **6**(13).

as.im.bivden *Converting a* sparr *bivariate kernel density estimate or relative risk surface object into a* spatstat *pixel image.*

Description

as.im methods for classes "bivden" and "rrs"

Usage

```
## S3 method for class 'bivden'
as.im(X, ...)
## S3 method for class 'rrs'
as.im(X, ...)
```

Arguments

X	An object of class "bivden" resulting from a call to bivariate.density, or an object of class "rss" resulting from a call to risk.
...	Ignored.

Value

An object of class im corresponding to the supplied argument. Additional return information originally part of X is lost.

Author(s)

T.M. Davies

See Also

im, as.im

Examples

```
data(chorley)

ch.bivden <- bivariate.density(chorley, ID = "lung", adaptive = FALSE, pilotH = 1.5)
ch.im <- as.im(ch.bivden)
summary(ch.im)
```

`bivariate.density` *Bivariate kernel density estimates*

Description

Provides an adaptive or fixed bandwidth kernel density estimate of bivariate data.

Usage

```
bivariate.density(data, ID = NULL, pilotH, globalH = pilotH,
  adaptive = TRUE, edgeCorrect = TRUE, res = 50, WIN = NULL,
  counts = NULL, intensity = FALSE, xrange = NULL,
  yrange = NULL, trim = 5, gamma = NULL, pdef = NULL,
  atExtraCoords = NULL, use.ppp.methods = TRUE, comment = TRUE)
```

Arguments

`data` An object of type `data.frame`, `list`, `matrix`, or `ppp` giving the observed data from which we wish to calculate the density estimate. Optional ID information (e.g. a dichotomous indicator for cases and controls) may also be provided in these four data structures. See ‘Details’ for further information on how to properly specify each one.

`ID` If `data` is a data structure with a third component/column indicating case (1) or control (0) status, `ID` must specify which of these groups we wish to estimate a density for. If `ID` is `NULL` (default), a density is estimated for all present observations, regardless of any status information.

`pilotH` A single numeric, positive ‘smoothing parameter’ or ‘bandwidth’. When `adaptive` is `TRUE` (default), this value is taken to be the pilot bandwidth, used to construct the bivariate pilot density required for adaptive smoothing (see ‘Details’). For a fixed bandwidth kernel density estimate, `pilotH` simply represents the fixed amount of smoothing. Currently, all smoothing is isotropic in nature.

`globalH` A single numeric, positive smoothing multiplier referred to as the global bandwidth, used to calculate the adaptive bandwidths (see ‘Details’). When `adaptive` is `TRUE`, this defaults to be the same as the pilot bandwidth. Ignored for a fixed density estimate.

`adaptive` Boolean. Whether or not to produce an adaptive (variable bandwidth) density estimate, with the alternative being a fixed bandwith density estimate. Defaults to `TRUE`.

`edgeCorrect` Boolean. Whether or not to perform edge-correction on the density estimate according to the methods demonstrated by Diggle (1985) (fixed bandwidth) and Marshall and Hazelton (2010) (adaptive). This can have a noticable effect on computation time in some cases. Defaults to `TRUE`. When `adaptive = TRUE`, the fixed-bandwidth pilot density is also edge-corrected according to `edgeCorrect`.

`res` A single, numeric, positive integer indicating the square root of the desired resolution of the evaluation grid. That is, each of the evaluation grid axes will have length `res`. Currently, only `res*res` grids are supported. Defaults to 50 for computational reasons.

`WIN` A polygonal object of class `owin` from the package `spatstat` giving the study region or ‘window’. All functions in the package `sparr` that require knowledge of the specific study region make use of this class; no other method of defining the study region is currently supported. If no window is supplied (default), the function defines (and returns) it’s own rectangular `owin` based on `xrange` and `yrange`. Ignored if `data` is an object of type `ppp`.

`counts` To perform binned kernel estimation, a numeric, positive, integer vector of giving counts associated with each observed coordinate in `data`, if `data` contains unique observations. If `NULL` (default), the function assumes each coordinate in `data` corresponds to one observation at that point. Should the `data` being supplied to `bivariate.density` contain duplicated coordinates, the function computes the counts vector internally (overriding any supplied value for `counts`), issues a warning, and continues with binned estimation. Non-integer values are rounded to the nearest integer.

`intensity` A boolean value indicating whether or not to return an intensity (interpreted as the the expected number of observations per unit area and integrating to the number of observations in the study region) function, rather than a density (integrating to one). Defaults to `FALSE`.

`xrange` Required only when no study region is supplied (`WIN = NULL`) and `data` is not an object of class `ppp`, and ignored otherwise. A vector of length 2 giving the upper and lower limits of the estimation interval for the x axis, in which case an evenly spaced set of values of length `res` is generated.

`yrange` As above, but for the y axis.

`trim` A numeric value (defaulting to 5) that prevents excessively large bandwidths in adaptive smoothing by trimming the originally computed bandwidths **h** by `trim` times `median(`**h**`)`. A value of `NA` or a negative numeric value requests no trimming. Ignored when `adaptive` is `FALSE`.

`gamma` An optional positive numeric value to use in place of `gamma` for adaptive bandwidth calculation (see ‘Details’). For adaptive relative risk estimation, this value can sensibly be chosen as common for both case and control densities (such as the `gamma` value from the adaptive density estimate of the ‘pooled’ (full) dataset) - see Davies and Hazelton (2010). If nothing is supplied (default), this value is computed from the data being used to estimate the density in the defined fashion (again, see ‘Details’). Ignored for fixed bandwidth estimation.

`pdef` An optional object of class `bivden` for adaptive density estimation. This object is used as an alternative or ‘external’ way to specify the pilot density for computing the variable bandwidth factors and must have the same grid resolution and coordinates as the estimate currently being constructed. If `NULL` (default) the pilot density is computed internally using `pilotH` from above, but if supplied, `pilotH` need not be given. Bandwidth trimming value is computed based upon the data points making up `pdef`. Ignored if `adaptive = FALSE`.

`atExtraCoords` It can occasionally be useful to retrieve the values of the estimated density at specific coordinates that are not the specific observations or the exact grid coordinates, for further analysis or plotting. `atExtraCoords` allows the user to specify an additional object of type `data.frame` with 2 colums giving the x `atExtraCoords[,1]` and y `atExtraCoords[,2]` coordinates at which to calculate and return the estimated density and other statistics (see 'Value').

`use.ppp.methods`

Boolean. Whether or not to switch to using methods defined for objects of class `ppp.object` from the package `spatstat` to estimate the density. This approach is much, much faster than forcing `bivariate.density` to do the explicit calculations (due to implementation of a Fast Fourier Transform; see `density.ppp`) and is highly recommended for large datasets. To further reduce computation time in the adaptive case when `use.ppp.methods = TRUE`, the variable edge-correction factors are calculated using the integer percentiles of the varying bandwidths. Defaults to `TRUE`.

`comment` Boolean. Whether or not to print function progress (including starting and ending times) during execution. Defaults to `TRUE`.

Details

This function calculates an adaptive or fixed bandwidth bivariate kernel density estimate, using the bivariate Gaussian kernel. Abramson's method is used for adaptive smoothing (Abramson, 1982). Suppose our `data` argumnent is a `data.frame` or `matrix`. Then for each observation `data[i,1:2]` (*i* = 1, 2, ... *n*), the bandwidth `h[i]` is given by

`h[i]=globalH` / (*w*(`data[i,1:2]`; `pilotH`)^(1/2)*`gamma`)

where *w* is the fixed bandwidth pilot density constructed with bandwidth `pilotH` and the scaling parameter `gamma` is the geometric mean of the *w*^(-1/2) values. A detailed discussion on this construction is given in Silverman (1986).

If the `data` argument is a `data.frame` or a `matrix`, this must have exactly two columns containing the x (`[,1]`) and y (`[,2]`) data values, or exactly three columns with the third (rightmost) column giving ID information by way of a numeric, dichotomous indicator. Should `data` be a `list`, this must have two vector components of equal length named `x` and `y`. The user may specify a third component with the name `ID` giving the vector of corresponding ID information (must be of equal length to `x` and `y`). Alternatively, `data` may be an object of class `ppp` (see `ppp.object`). ID information can be stored in such an object through the argument `marks`. If `data` is a `ppp` object, the value of `window` of this object overrides the value of the argument `WIN` above.

Value

An object of class `"bivden"`. This is effectively a list with the following components:

`Zm` a numeric matrix giving the value of the estimated (edge-corrected if elected) density at each of the coordinates of the grid. Values corresponding to points on the grid that fall outside the study region `WIN` are set to `NA`

`X` a the sequence of values that were used as x grid coordinates. Will have length `res`

Y a the sequence of values that were used as y grid coordinates. Will have length res

kType the kernel function used in estimation. Currently fixed at "gaus"

h a numeric vector with length equal to the number of observations, giving the bandwidths assigned to each observation in the order they appeared in data. For a fixed bandwidth estimate, this will simply be the identical value passed to and returned as pilotH

pilotH the pilot or fixed bandwidth depending on whether adaptive smoothing is employed or not, respectively

globalH the global bandwidth globalH if adaptive smoothing is employed, NA for fixed smoothing

hypoH the matrix of 'hypothetical' bandwidths (with element placement corresponding to Zm) for each coordinate of the evaluation grid. That is, these values are the bandwidths at that grid coordinate if, hypothetically, there was an observation there (along with the original data). These are used for edge-correction in adaptive densities (Marshall and Hazelton, 2010). Will be NA for fixed bandwidth estimates

zSpec a numeric vector with length equal to the number of observations used, giving the values of the density at the specific coordinates of the observations. Order corresponds to the order of the observations in data

zExtra as zSpec for the observations in atExtraCoords, NA if atExtraCoords is not supplied

WIN the object of class owin used as the study region

qhz a numeric matrix of the edge-correction factors for the entire evaluation grid (with placement corresponding to Zm. If edgeCorrect = FALSE, all edge correction factors are set to and returned as 1

qhzSpec edge-correction factors for the individual observations; order corresponding to data

qhzExtra as qhzSpec for the observations in atExtraCoords; NA if atExtraCoords is not supplied

pdef a copy of the object originally supplied as the pilot density as per the pdef argument; NULL if unused

pilotvals the values of the pilot density used to compute the adaptive bandwidths. Order corresponds to the order of the observations in data. NULL when adaptive = FALSE

gamma the value of gamma that was passed to the function, or the geometric mean term of the reciprocal of the square root of the pilot density values used to scale the adaptive bandwidths if gamma is not supplied. NULL when adaptive = FALSE

counts the counts vector used in estimation of the density/intensity. If all values in data were unique and counts = NULL, the returned counts will be a vector of ones equal to the number of coordinates in data

data a two-column numeric data frame giving the observations in the originally supplied data that were used for the density estimation. If data originally contained duplicated coordinates, the returned data will contain only the unique coordinates, and should be viewed with respect to the returned value of counts

Warning

Explicit calculation of bivariate kernel density estimates is computationally expensive. The decision to produce adaptive over fixed bandwidth estimates, the size of the dataset, the evaluation grid resolution specified by `res`, the complexity of the study region and electing to edge-correct all have a direct impact upon the time it will take to estimate the density. Keeping `use.ppp.methods = TRUE` can drastically reduce this computational cost at the expense a degree of accuracy that is generally considered negligible for most practical purposes.

Author(s)

T.M. Davies

References

Abramson, I. (1982). On bandwidth variation in kernel estimates — a square root law, *Annals of Statistics*, **10**(4), 1217-1223.

Davies, T.M. and Hazelton, M.L. (2010), Adaptive kernel estimation of spatial relative risk, *Statistics in Medicine*, **29**(23) 2423-2437.

Davies, T.M., Jones, K. and Hazelton, M.L. (2015), Symmetric adaptive smoothing regimens for estimation of the spatial relative risk function, *Submitted for publication*.

Diggle, P.J. (1985), A kernel method for smoothing point process data, *Journal of the Royal Statistical Society, Series C*, **34**(2), 138-147.

Marshall, J.C. and Hazelton, M.L. (2010) Boundary kernels for adaptive density estimators on regions with irregular boundaries, *Journal of Multivariate Analysis*, **101**, 949-963.

Silverman, B.W. (1986), *Density Estimation for Statistics and Data Analysis*, Chapman & Hall, New York.

Examples

```
##Chorley-Ribble laryngeal cancer data ('spatstat' library)
data(chorley)

ch.lar.density <- bivariate.density(data = chorley, ID = "larynx",
 pilotH = 1.5, adaptive = FALSE)

plot(ch.lar.density, col = "lightblue", phi = 30, theta = -30,
 ticktype = "detailed", main = "chorley.larynx", display = "persp")

## Not run:

##PBC liver disease data
data(PBC)
pbc.adaptive.density <- bivariate.density(data = PBC, ID = "case",
 pilotH = 350)

#3D plot - may need to adjust size of RGL device. Hold left click
# to pan, hold right to zoom.
plot(pbc.adaptive.density, display = "3d", col = heat.colors(20),
 main = "Density of PBC in north-east England", aspect = 1:2)

## End(Not run)
```

KBivN *Standard bivariate normal kernel*

Description

Evaluates the standard bivariate normal (Gaussian) kernel function at specified values.

Usage

```
KBivN(X)
```

Arguments

X	A numeric vector of length 2 or a data frame with 2 columns.

Details

If X is a vector of length 2, then the two components X[1] and X[2] are taken to be the x and y coordinates respectively. For multiple evaluations at differing coordinates, X must be a data frame with X[,1] and X[,2] as the corresponding pairs of x and y coordinates respectively.

Value

A single numeric value if X is a vector, or nrow(X) values if X is a data frame, giving the result of the standard bivariate normal kernel at the specified coordinate(s).

Author(s)

T.M. Davies

Examples

```
KBivN(c(0.1,0.4))

x <- y <- seq(-4,4,length=50)
z <- KBivN(data.frame(cbind(sort(rep(x,50)),rep(y,50))))
persp(x,y,matrix(z,50,50,byrow=TRUE),main="bivariate Gaussian kernel",
 phi=30,theta=-30)
```

`KBivQ` *Standard bivariate quartic (biweight) kernel*

Description

Evaluates the standard bivariate quartic (biweight) kernel function at specified values, for either the spherical or product derivation of the function.

Usage

```
KBivQ(X,type="spher")
```

Arguments

`X`	A numeric vector of length 2 or a data frame with 2 columns.
`type`	A character string.

`"spher"`	(default) selects spherical method of calculating the bivariate quartic kernel function
`"prod"`	uses the product approach to calculating the function

Details

If `X` is a vector of length 2, then the two components `X[1]` and `X[2]` are taken to be the x and y coordinates respectively. For multiple evaluations at differing coordinates, `X` must be a data frame with `X[,1]` and `X[,2]` as the corresponding pairs of x and y coordinates respectively.

Unlike the bivariate Gaussian kernel, it is necessary to specify the method of extending the univariate quartic kernel to the bivariate case; this can be done in two different ways, one way resulting in a slightly different kernel to the other. An explanation of these ‘spherical’ and ‘product’ approaches is given in Wand and Jones (1995).

Value

A single numeric value if `X` is a vector, or `nrow(X)` values if `X` is a data frame, giving the result of the standard bivariate quartic kernel at the specified coordinate(s) for the elected function derivation type.

Author(s)

T.M. Davies

References

Wand, M.P. and Jones, C.M. (1995), *Kernel Smoothing*, Chapman & Hall, London.

Examples

```
KBivQ(c(0.1,0.4))
```

```
x <- y <- seq(-0.9,0.9,length=50)
z.spher <- KBivQ(data.frame(cbind(sort(rep(x,50)), rep(y,50))))
z.prod <- KBivQ(data.frame(cbind(sort(rep(x,50)), rep(y,50))), "prod")

par(mfrow=c(1,2))
contour(x, y, matrix(z.spher, 50, 50, byrow = TRUE),
 main = "bivariate quartic kernel (spherical)")
contour(x, y, matrix(z.prod, 50, 50, byrow=TRUE),
 main = "bivariate quartic kernel (product)")
```

LSCV.density   *Leave-one-out least-squares cross-validation (LSCV) for bivariate KDE bandwidths*

Description

Provides an isotropic LSCV bandwidth estimate for use in 2-dimensional kernel density estimation (see e.g. Bowman and Azzalini, 1997).

Usage

```
LSCV.density(data, hlim = NULL, res = 128, edge = TRUE,
 WIN = NULL, quick = TRUE, comment = TRUE)
```

Arguments

`data` An object of type `data.frame`, `list`, `matrix`, or `ppp` describing the observed data from which we wish to calculate the LSCV bandwidth. See ‘Details’ for further information.

`hlim` A numeric vector of length 2 giving the interval over which to search for the bandwidth that minimises the selection criterion. If `NULL` (default), the function attempts to automatically select an appropriate range based on multiples of Stoyan and Stoyan’s (1994) rule-of-thumb. The user is strongly recommended to supply their own `hlim`.

`res` Single integer giving the square grid resolution over which evaluation of the selection criterion takes place. Defaults to a 128 by 128 grid.

`edge` Boolean. Whether or not to employ edge-correction in the calculations. Defaults to `TRUE`.

`WIN` A polygonal `owin` object giving the study region. Ignored if `data` is already a `ppp.object`.

`quick` Intended for advanced use; users are recommended not to change the default `TRUE`. Setting `quick = FALSE` forces the function to individually evaluate the CV objective function at each of `seq(hlim[1], hlim[2], length = 50)` bandwidths, returning the corresponding values. Can be useful for diagnostic purposes.

comment Boolean. Whether or not to print function progress during execution. Defaults to TRUE.

Details

This function calculates a LSCV smoothing bandwidth for kernel density estimates of 2-dimensional (bivariate) data. If the data argument is a data.frame or a matrix, this must have exactly two columns containing the x ([,1]) and y ([,2]) data values. Should data be a list, this must have two vector components of equal length named x and y. Alternatively, data may be an object of class ppp (see ppp.object).

Value

A single numeric value of the estimated bandwidth (if quick = FALSE, this value is named hopt; additionally returned are the objective function values (lscv) and the index of the minimum value (ind)). The user may need to experiment with adjusting hlim to find a suitable minimum.

Warning

Leave-one-out LSCV for bandwidth selection in kernel density estimation is notoriously unstable in practice and has a tendency to produce rather small bandwidths. Satisfactory bandwidths are not guaranteed for every application. This method can also be computationally expensive for large data sets and fine evaluation grid resolutions.

Author(s)

T.M. Davies

References

Bowman, A.W. and Azzalini, A. (1997), *Applied Smoothing Techniques for Data Analysis: The Kernel Approach with S-Plus Illustrations.* Oxford University Press Inc., New York. ISBN 0-19-852396-3.

Stoyan, D. and Stoyan, H. (1994), *Fractals, Random Shapes and Point Fields.* Wiley, Great Britain. ISBN 0-471-93757-6.

See Also

spatstat's function bw.relrisk

Examples

```
## Not run:
data(PBC)

##PBC cases
LSCV.density(split(PBC)[[1]],hlim=c(10,400))
```

```
##PBC controls
LSCV.density(split(PBC)[[2]],hlim=c(10,400))

## End(Not run)
```

`LSCV.risk` *Leave-one-out least-squares cross-validation (LSCV) bandwidths for the relative risk function*

Description

Attempts to estimate a jointly optimal, common case-control fixed bandwidth for use in the kernel-smoothed relative risk function via leave-one-out least-squares cross-validation (LSCV). The user can choose between two methods described in Kelsall and Diggle (1995a;b) and Hazelton (2008).

Usage

```
LSCV.risk(cases, controls, hlim = NULL,
 method = c("kelsall-diggle", "hazelton"), res = 128,
 WIN = NULL, edge = TRUE, comment = TRUE)
```

Arguments

`cases` An object of type `data.frame`, `list`, `matrix`, or `ppp` describing the observed case data from which we wish to calculate the LSCV bandwidth. See 'Details' for further information.

`controls` As for `cases`, but for the control observations. Both `cases` and `controls` must be of the same object class.

`hlim` A numeric vector of length 2 giving the interval over which to search for the common bandwidth which minimises the selection criterion. If `NULL` (default), the function attempts to automatically select an appropriate range based on multiples of Stoyan and Stoyan's (1994) rule-of-thumb. The user is strongly recommended to supply their own `hlim`.

`method` A character vector giving the specific selection criterion to minimise; see either Kelsall and Diggle (1995b) or Hazelton (2008). See 'Details'. Defaults to `"kelsall-diggle"`.

`res` Single integer giving the square grid resolution over which evaluation of the selection criterion takes place. Defaults to a 128 by 128 grid.

`WIN` A polygonal `owin` object giving the study region. Ignored if `data` is already a `ppp.object`.

`edge` Boolean. Whether or not to employ edge-correction in the calculations. Defaults to `TRUE`.

`comment` Boolean. Whether or not to print function progress during execution. Defaults to `TRUE`.

Details

This function calculates a 'jointly optimal', common isotropic LSCV bandwidth for the (Gaussian) kernel-smoothed relative risk function (case-control density-ratio). If the `cases`, `controls` arguments are `data.frame` or `matrix` objects, these must each have exactly two columns containing the `x` (`[,1]`) and `y` (`[,2]`) data values. Should they be `list`s, these must have two vector components of equal length named `x` and `y`. Alternatively, `cases` and `controls` may be objects of class `ppp` (see `ppp.object`), and the argument `WIN` can be ignored.

It can be shown that choosing a bandwidth that is equal for both case and control density estimates is preferable to computing 'separately optimal' bandwidths (Kelsall and Diggle, 1995a). Setting `method = "kelsall-diggle"`, `LSCV.risk` computes the common bandwidth which minimises the approximate mean integrated squared error of the log-transformed risk surface (see specifically Kelsall and Diggle, 1995b).

Alternatively, the user has the option of computing the common case-control bandwidth which minimises a *weighted* mean integrated squared error of the (raw) relative risk function (see Hazelton, 2008). Generally, this author has found the Kelsall-Diggle method to provide more stable performance.

Value

A single numeric value of the estimated bandwidth. The user may need to experiment with adjusting `hlim` to find a suitable minimum.

Warning

Leave-one-out LSCV for jointly optimal, common bandwidth selection in the kernel-smoothed risk function is even more unstable (in terms of high variability) than the standalone density version. Caution is advised; not all applications will yield a successful result (this is termed "a breakdown of the methodology" by Kelsall and Diggle, 1995a). Undersmoothing has been noted in this author's personal experience. This method can also be computationally expensive for large data sets and fine evaluation grid resolutions.

Author(s)

T.M. Davies

References

Kelsall, J.E. and Diggle, P.J. (1995a), Kernel estimation of relative risk, *Bernoulli*, **1**, 3-16.

Kelsall, J.E. and Diggle, P.J. (1995b), Non-parametric estimation of spatial variation in relative risk, *Statistics in Medicine*, **14**, 2335-2342.

Hazelton, M. L. (2008), Letter to the editor: Kernel estimation of risk surfaces without the need for edge correction, *Statistics in Medicine*, **27**, 2269-2272.

Stoyan, D. and Stoyan, H. (1994), *Fractals, Random Shapes and Point Fields.* Wiley, Great Britain. ISBN 0-471-93757-6.

See Also

spatstat's function bw.relrisk

Examples

```
## Not run:
data(chorley)

LSCV.risk(cases = split(chorley)[[1]], controls = split(chorley)[[2]],
 hlim = c(0.1,2))

## End(Not run)
```

NS *Normal scale rule for bivariate KDE bandwidths*

Description

Provides the (isotropic) optimal bandwidth for a bivariate normal density based on a simple expression.

Usage

```
NS(data, nstar = NULL, scaler = NA)
```

Arguments

data	An object of type data.frame, list, matrix, or ppp giving the observed data from which we wish to calculate the NS bandwidth. See 'Details' for further information.
nstar	A single numeric, positive value to use in place of the number of observations n in the NS formula. If NULL (default), n will simply be the number of observations in data.
scaler	A single numeric, positive value to use for transforming the result with respect to the scale of the recorded data (i.e. a scalar representation of the standard deviation of the data). If NA (default), the scaling value is set as the mean of the interquartile ranges (IQR) of the x and y data values divided by 1.34 (Gaussian IQR).

Details

This function calculates a smoothing bandwidth for kernel density estimates of 2-dimensional data: the optimal value which would minimise the asymptotic mean integrated squared error of the bivariate normal density function, assuming the standard Gaussian kernel function. See Wand and Jones (1995) for example. If the `data` argument is a `data.frame` or a `matrix`, this must have exactly two columns containing the *x* (`[,1]`) and *y* (`[,2]`) data values. Should `data` be a `list`, this must have two vector components of equal length named `x` and `y`. Alternatively, `data` may be an object of class `ppp` (see `ppp.object`).

Value

A single numeric value of the estimated bandwidth.

Warning

The NS bandwidth is an approximation, and assumes *that the target density is bivariate normal*. This is considered rare in e.g. epidemiological applications. Nevertheless, it remains a quick and easy ‘rule-of-thumb’ method with which one may obtain a smoothing parameter in general applications.

Author(s)

T.M. Davies

References

Wand, M.P. and Jones, C.M., 1995. *Kernel Smoothing*, Chapman & Hall, London.

Examples

```
data(PBC)
PBC.casedata <- split(PBC)[[1]]
PBC.controldata <- split(PBC)[[2]]

pbc.h <- NS(PBC, nstar = sqrt(PBC.casedata$n * PBC.controldata$n))


##Scaling bandwidth for case data using standard deviations
sd.x <- sd(PBC.casedata$x)
sd.y <- sd(PBC.casedata$y)

NS(data = PBC.casedata, scaler = mean(sd.x, sd.y))
```

OS *Maximal smoothing principle (oversmoothing) for bivariate KDE bandwidths*

Description

Provides an (isotropic) bandwidth estimate for use in bivariate KDE based on the oversmoothing factor introduced by Terrell (1990).

Usage

```
OS(data, nstar = NULL, scaler = NA)
```

Arguments

`data`	An object of type `data.frame`, `list`, `matrix`, or `ppp` giving the observed data from which we wish to calculate the OS bandwidth. See 'Details' for further information.
`nstar`	A single numeric, positive value to use in place of the number of observations *n* in the OS formula. If `NULL` (default), *n* will simply be the number of observations in `data`.
`scaler`	A single numeric, positive value to use for transforming the result with respect to the scale of the recorded data. If `NA` (default), the scaling value is set as the mean of the interquartile ranges (IQR) of the *x* and *y* data values divided by 1.34 (Gaussian IQR). This approach was used in Davies and Hazelton (2010).

Details

This function calculates a smoothing bandwidth for kernel density estimates of bivariate data, following the maximal smoothing priciple of Terrell (1990). If the `data` argument is a `data.frame` or a `matrix`, this must have exactly two columns containing the *x* (`[,1]`) and *y* (`[,2]`) data values. Should `data` be a `list`, this must have two vector components of equal length named `x` and `y`. Alternatively, `data` may be an object of class `ppp` (see `ppp.object`).

Value

A single numeric value of the estimated bandwidth.

Author(s)

T.M. Davies

References

Davies, T.M. and Hazelton, M.L. (2010), Adaptive kernel estimation of spatial relative risk, *Statistics in Medicine*, **29**(23) 2423-2437.

Terrell, G.R. (1990), The maximal smoothing principle in density estimation, *Journal of the American Statistical Association*, **85**, 470-477.

Examples

```
data(PBC)
PBC.casedata <- split(PBC)[[1]]
PBC.controldata <- split(PBC)[[2]]

pbc.h <- OS(PBC, nstar = sqrt(PBC.casedata$n * PBC.controldata$n))


##Scaling bandwidth for case data using standard deviations
sd.x <- sd(PBC.casedata$x)
sd.y <- sd(PBC.casedata$y)

OS(data = PBC.casedata, scaler = mean(sd.x, sd.y))
```

PBC *Primary biliary cirrhosis data*

Description

Data of the locations of 761 cases of primary biliary cirrhosis in several adjacent health regions of north-eastern England, along with 3020 controls representing the at-risk population, collected between 1987 and 1994. These data were first presented and analysed by Prince et al. (2001); subsequent analysis of these data in the spirit of sparr was performed in Davies and Hazelton (2010). Also included is the polygonal study region.

Usage

```
data(PBC)
```

Format

PBC is a dichotomously marked ppp.object.

Acknowledgements

The authors thank Prof. Peter Diggle at Lancaster University (http://www.lancs.ac.uk/staff/diggle/) for providing access to these data.

Source

Prince et al. (2001), The geographical distribution of primary biliary cirrhosis in a well-defined cohort, *Hepatology*, **34**, 1083-1088.

References

Davies, T.M. and Hazelton, M.L. (2010), Adaptive kernel estimation of spatial relative risk, *Statistics in Medicine*, **29**(23) 2423-2437.

Examples

```
data(PBC)
summary(PBC)
plot(PBC)
```

---

plot.bivden *Plotting a bivariate kernel density estimate object*

Description

plot methods for classes "bivden" and "rrs"

Usage

```
## S3 method for class 'bivden'
plot(x, ..., display = c("heat", "contour", "persp", "3d"),
 show.WIN = TRUE)
## S3 method for class 'rrs'
plot(x, ..., display = c("heat", "contour", "persp", "3d"),
 show.WIN = TRUE, tolerance.matrix = NULL,
 tol.opt = list(raise = 0.01, col = "black", levels = 0.05, lty = 1, lwd = 1))
```

Arguments

x	An object of class "bivden" resulting from a call to bivariate.density, or an object of class "rss" resulting from a call to risk.
...	Additional graphical parameters to be passed to the relevant plot command depending on the value of display.
display	One of four possible character strings indicating the kind of plot desired (see 'Details'). Defaults to "heat".
show.WIN	Boolean. Whether or not to draw the study region as an aesthetic enhancement to the plot of the density/risk surface. Defaults to TRUE.

tolerance.matrix
: The matrix of p-values resulting from a call to tolerance and used to draw the asymptotic tolerance contours. If this argument is supplied, tolerance contours are automatically superimposed upon a display = "heat" or display = "3d" plot. Ignored for display = "persp" or display = "contour" plots. Defaults to NULL.

tol.opt
: A named list of components that control plotting of the tolerance contours given by tolerance.matrix. Components col, levels, lty, and lwd are vectors of equal length controlling the colour, significance levels, line type (ignored for display = "3d") and line width of the plotted contours respectively. The element raise is a single numeric value and is used only when display = "3d". This vertically (i.e. with respect to the z axis) translates the contours atop the 3-D surface (see 'Details'). A value of 0 requests no translation. Defaults to 0.01.

Details

There are currently four implemented plot types to visualise the estimated density or risk function. "heat" selects a heatplot, "contour" is simply a contour plot and "persp" creates a perspective plot. Selection of "3d" uses functions from the rgl package to open an RGL graphics device and creates a 3-dimensional surface which the user can interact with using the mouse. To use ... to improve the appearance of the four possible plot types "heat", "contour", "persp" and "3d", the reader is highly recommended to consult the relevant documentation in the help pages plot.im, contour, persp and persp3d respectively.

Adding tolerance contours to a "3d" relative risk plot requires the function to make some approximations to the vertical positioning of the contours at each corresponding coordinate. This can lead to some parts of normally visible contours falling 'underneath' the plotted surface, resulting in partially obscured contours. The element raise in tol.opt overcomes this issue by artificially raising the visible contours by a fixed amount. Care should be taken to find an appropriate value for raise for each analysis.

Value

Plots to the relevant graphics device.

Author(s)

T.M. Davies

See Also

bivariate.density, risk, plot.default, plot.im, contour,
persp, persp3d, par, par3d

Examples

```
## see Examples in documentation for functions 'bivariate.density',
##  'risk' and 'tolerance'.
```

`risk` *Bivariate relative risk function*

Description

Estimates a *relative risk function* based on the ratio of two bivariate kernel density estimates over identical grids and regions. In geographical epidemiology, the two densities would represent a set of disease cases (numerator) and a sample of controls illustrating the at-risk population (denominator). In epidemiological terminology, the ratio of 'case' to 'control' would technically be referred to as an *odds ratio*.

Usage

```
risk(f, g, delta = 0, log = TRUE, h = NULL, adaptive = FALSE, res = 50,
 WIN = NULL, tolerate = FALSE, plotit = TRUE, comment = TRUE)
```

Arguments

`f` Either a pre-calculated object of class `"bivden"` representing the 'case' density estimate, or an object of type `data.frame`, `list`, `matrix`, or `ppp` giving the observed case data. If this raw data is provided, a kernel density estimate is computed internally, with certain options available to the user in `bivariate.density` chosen/calculated automatically. See 'Details' for further information.

`g` As for argument `f`, but for the controls. Whatever the type, the class of `g` must match that of `f`.

`delta` A single numeric scaling parameter used for an optional additive constant to the densities; occasionally used for risk surface construction (see 'Details'). A negative or zero value for `delta` requests no additive constant (default).

`log` Boolean. Whether or not to return the (natural) log-transformed relative risk function as recommended by Kelsall and Diggle (1995a). Defaults to `TRUE` with the alternative being the raw density ratio.

`h` Ignored if `f` and `g` are already `"bivden"` objects. An optional numeric vector of length 1 OR 2, giving the global bandwidth(s) for internal estimation of the case and control densities if `adaptive = TRUE`, or the fixed bandwidth(s) if `adapative = FALSE`. When `h` is a single numeric value, this is elected as the common global/fixed bandwidth for case and control densities. When `h` has length 2, the values `h[1]` and `h[2]` are assigned as the case and control global/fixed bandwidths respectively. By default, a value of `h = NULL` tells the function to use the global/fixed smoothing parameters as outlined in 'Details' below. Note that for adaptive estimation, this argument does not affect calculation of the pilot bandwiths.

`adaptive` Ignored if `f` and `g` are already `"bivden"` objects. A boolean value specifying whether or not to employ adaptive smoothing for internally estimating the densities. A value of `FALSE` (default) elects use of fixed-bandwidth estimates.

`res` Ignored if `f` and `g` are already `"bivden"` objects. A numeric value giving the desired resolution (of one side) of the evaluation grid. Higher values increase resolution at the expense of computational efficiency. Defaults to a 50 by 50 grid.

`WIN` Ignored if `f` and `g` are already `"bivden"` objects OR objects of class `ppp` (in which case the study region is set to the value of the resident `window` component). A polygonal object of class `owin` giving the relevant study region in which the `f` and `g` data was collected.

`tolerate` Ignored if `f` and `g` are already `"bivden"` objects. A boolean value specifying whether or not to calculate a corresponding asymptotic p-value surface (for tolerance contours) for the estimated relative risk function. If `TRUE`, the p-value surface tests for elevated risk only (equivalent to setting `test = "greater"` in `tolerance`) and is evaluated over a maximum grid resolution of 50 by 50. Defaults to `FALSE` for computational reasons.

`plotit` Boolean. If `TRUE` (default), a heatplot of the estimated relative risk function is produced. If `tolerate = TRUE`, asymptotic tolerance contours are automatically added to the plot at a significance level of 5%.

`comment` Ignored if `f` and `g` are already `"bivden"` objects. Boolean. Whether or not to print function progress (including starting and ending date-times) during execution. Defaults to `TRUE`.

Details

This function estimates a relative risk function via the density ratio method using fixed or adaptive bandwidth bivariate kernel density estimates. Both densities must be estimated using the same evaluation grid (and the same study window) in `bivariate.density`. In geographical epidemiology, the argument `f` represents the spatial distribution of the disease cases, and `g` the at-risk (control) population.

The option to supply the raw case and control data is available. If this is done, the function runs `bivariate.density` internally, abstracting certain decisions about the density estimation away from the user. If the user sets `adaptive = TRUE` (and `h` remains at `NULL`), the smoothing parameters are calculated as per the approach taken in Davies and Hazelton (2010): a common global bandwidth using the pooled data from `OS`. Pilot bandwidths are set at half the corresponding `OS` values. The scaling parameter `gamma` is common for the case and control density estimates, set as the `gamma` component of the pooled estimate. If a fixed relative risk is desired (`adaptive = FALSE`) and no specific bandwidths are given via the argument `h`, the case and control densities share a common bandwidth computed from the pooled data using `OS`. In supplying raw data to `risk`, the user must also specify an evaluation grid resolution (defaulting to 50 by 50) and the study region `WIN` (unless `f` and `g` are objects of class `ppp`, in which case the resident `window` component overrides `WIN`). All other arguments are set to their defaults as in `bivariate.density`.

If more flexibility is required for estimation of the case and control densities, the user must supply ‘pre-calculated’ objects of class `"bivden"` (from `bivariate.density`) as the `f` and `g` arguments. This drastically reduces the running time of a call to `risk` (as the density estimation step is already complete). However, the option of internally computing the asymptotic p-value surfaces (via the argument `tolerate`) is unavailable in this case; the user must run the `tolerance` function separately if tolerance contours are desired.

The relative risk function is defined here as the ratio of the 'case' density to the 'control' (Bithell, 1990; 1991). Using kernel density estimation to model these densities (Diggle, 1985), we obtain a workable estimate thereof. This function defines the risk function *r* in the following fashion:

r `= (f + delta*max(g))/(g + delta*max(g))`

Note the (optional) additive constants defined by `delta` times the maximum of each of the densities in the numerator and denominator respectively (see Bowman and Azzalini, 1997).

The log-risk function *rho*, given by *rho* = log[*r*], is argued to be preferable in practice as it imparts a sense of symmetry in the way the case and control densities are treated (Kelsall and Diggle, 1995a;b). The option of log-transforming the returned risk function is therefore selected by default.

Value

An object of class `"rrs"`. This is a marked list with the following components:

`rsM` a numeric `res*res` matrix (where `res` is the grid resolution as specified in the calls to `bivariate.density` for calculation of `f` and `g`) giving the values of the risk surface over the evaluation grid. Values corresponding to grid coordinates outside the study region are assigned `NA`

`f` the object of class `"bivden"` used as the numerator or 'case' density estimate

`g` the object of class `"bivden"` used as the denominator or 'control' density estimate

`log` whether or not the returned risk function is on the log-scale

`pooled` the object of class `"bivden"` (based on the pooled data) calculated internally if `f` and `g` were raw data arguments, `NA` otherwise

`P` a numeric 50 by 50 matrix of the asymptotic p-value surface if `tolerate = TRUE` and `f` and `g` were raw data arguments, `NA` otherwise

Warning

If raw data is supplied to `risk`, as opposed to previously computed objects of class `"bivden"`, the running time of this function will be greater. This is particularly the case if the user has also selected `tolerate = TRUE`. In the same fashion as `bivariate.density` and `tolerance`, setting `comment = TRUE` can keep the user appraised of the function progress during run-time.

Author(s)

T.M. Davies, M.L. Hazelton and J.C. Marshall

References

Bithell, J.F. (1990), An application of density estimation to geographical epidemiology, *Statistics in Medicine*, **9**, 691-701.

Bithell, J.F. (1991), Estimation of relative risk functions, *Statistics in Medicine*, **10**, 1745-1751.

Bowman, A.W. and Azzalini A. (1997), *Applied Smoothing Techniques for Data Analysis: The*

Kernel Approach with S-Plus Illustrations, Oxford University Press Inc., New York.

Davies, T.M. and Hazelton, M.L. (2010), Adaptive kernel estimation of spatial relative risk, *Statistics in Medicine*, **29**(23) 2423-2437.

Diggle, P.J. (1985), A kernel method for smoothing point process data, *Journal of the Royal Statistical Society Series C*, **34**(2), 138-147.

Kelsall, J.E. and Diggle, P.J. (1995a), Kernel estimation of relative risk, *Bernoulli*, **1**, 3-16.

Kelsall, J.E. and Diggle, P.J. (1995b), Non-parametric estimation of spatial variation in relative risk, *Statistics in Medicine*, **14**, 2335-2342.

Examples

```
## Not run:
data(PBC)
PBC.casedata <- split(PBC)[[1]]
PBC.controldata <- split(PBC)[[2]]

pbc.h <- OS(PBC, nstar = sqrt(PBC.casedata$n * PBC.controldata$n))

pbc.pool <- bivariate.density(data = PBC, pilotH = pbc.h,
 adaptive = FALSE)
pbc.case <- bivariate.density(data = PBC.casedata,
 pilotH = pbc.h, adaptive = FALSE)
pbc.con <- bivariate.density(data = PBC.controldata,
 pilotH = pbc.h, adaptive = FALSE)

pbc.rrs <- risk(f = pbc.case, g = pbc.con, plotit = FALSE)
summary(pbc.rrs)


## End(Not run)
```

summary.bivden *Summarising a bivariate kernel density estimate object*

Description

print and summary methods for class "bivden"

Usage

```
## S3 method for class 'bivden'
print(x, ...)
## S3 method for class 'bivden'
summary(object, ...)
```

Arguments

`x, object`	An object of class `"bivden"`.
`...`	Ignored.

Author(s)

T.M. Davies

`summary.rrs` *Summarising an estimated relative risk function object*

Description

`print` and `summary` methods for class `"rrs"`

Usage

```
## S3 method for class 'rrs'
print(x, ...)
## S3 method for class 'rrs'
summary(object, ...)
```

Arguments

`x, object`	An object of class `"rrs"` resulting from a call to `risk`.
`...`	Ignored.

Author(s)

T.M. Davies

`tolerance` *Asymptotic p-value surfaces*

Description

Calculates pointwise p-values based on asymptotic theory or Monte-Carlo (MC) permutations describing the extremity of risk over a given fixed or adaptive kernel-smoothed relative risk function.

Usage

```
tolerance(rs, test = "upper", method = "ASY",
    pooled = NULL, symchoice = NULL, hpsim = NULL,
    h0sim = NULL, reduce = 1, ITER = 1000,
    exactL2 = TRUE, comment = TRUE)
```

Arguments

rs An object of class "rrs" resulting from a call to risk, giving the fixed or adaptive kernel-smoothed risk function.

test A character string indicating the kind of test desired to yield the p-values. Must be one of "upper" (default - performs upper tailed tests examining heighted risk ‘hotspots’), "lower" (lower tailed tests examining ‘troughs’) or "double" (double-sided tests). See ‘Details’ for further information.

method A character string, either "ASY" (default) or "MC" indicating which method to use for calculating the p-value surface (asymptotic and Monte-Carlo approaches respectively). The MC approach is far more computationally expensive than the asymptotic method (see ‘Warnings’). See ‘Details’ for more information on the Monte-Carlo implementation.

pooled Required if method = "ASY", ignored otherwise. An object of class "bivden" resulting from a call to bivariate.density (or the component pooled from rs if it was created using raw data arguments) representing a density estimate based on the ‘pooled’ dataset of both ‘case’ and ‘control’ points. If separate from rs, this pooled density estimate must follow the same smoothing approach, evaluation grid and study region window as the densities used to create rs.

symchoice Currently only implemented for method = "MC"; ignored otherwise (asymptotic version still in development). A character string denoting the density to use as the common pilot estimate for the symmetric adaptive estimator (Davies et al. 2015). If symchoice = "f" the case density estimate is used, if symchoice = "g" it’s the control, and if symchoice = "pool" then a pooled estimate is used. If nothing is specified, i.e. symchoice is set at the default NULL, then the function implements the adaptive asymmetric estimator for the iterations. See ‘Details’.

hpsim Used to specify the pilot bandwidth(s) for the Monte-Carlo simulated tolerance contours. Can be NULL (default), in which case the pilot bandwidths used at each iteration are the same values used for the original density estimates used in creation of rs; a numeric vector of length 1 (pilot bandwidths for case and control densities are the same) or 2 (in the order of c(pilot f,pilot g)); or a function (see ‘Details’ for the latter). If the risk function in rs is a fixed-bandwidth estimate, then hpsim is used to directly specify the case and control density estimate bandwidths at each iteration. Ignored if method = "ASY".

h0sim As for hpsim, but for the global bandwidths used at each iteration when computing Monte-Carlo tolerance contours for adaptive estimates in rs. Not used if rs is a fixed-bandwidth estimate.

reduce A numeric value greater than zero and less than or equal to one giving the user the option to reduce the resolution of the evaluation grid for the pointwise p-values by specifying a proportion of the size of the evaluation grid for the original density estimates. For example, if the case and control "bivden" objects were calculated using res = 100 and tolerance was called with reduce = 0.5, the p-value surface will be evaluated over a 50 by 50 grid. A non-integer value resulting from use of reduce will be ceilinged.

ITER An integer value specifying the number of iterations to be used if method = "MC" (defaulting to 1000). Non-integer numeric values are rounded. Ignored when method = "ASY".

exactL2 Ignored if rs (and pooled) are fixed-bandwidth density estimates, or if method = "MC". A boolean value indicating whether or not to separately calculate the 'L2' integral components for adaptive tolerance contours. A value of FALSE will approximate these components based on the 'K2' integrals for faster execution (depending on the size of the evaluation grid, this improvement may be small) at the expense of a small degree of accuracy. Defaults to TRUE. See the reference for adaptive p-value surfaces in 'Details' for definitions of these integral components.

comment Boolean. Whether or not to print function progress (including starting and ending times) during execution. Defaults to TRUE.

Details

This function implements developments in Hazelton and Davies (2009) (fixed) and Davies and Hazelton (2010) (adaptive) to compute pointwise p-value surfaces based on asymptotic theory of kernel-smoothed relative risk surfaces. Alternatively, the user may elect to calculate the p-value surfaces using Monte-Carlo methods (see Kelsall and Diggle, 1995). Superimposing upon a plot of the risk surface contours of these p-values at given significance levels (i.e. 'tolerance contours') can be an informative way of exploring the statistical significance of the extremity of risk across the defined study region. The asymptotic approach to the p-value calculation is advantageous over a Monte-Carlo method, which can lead to excessive computation time for adaptive risk surfaces and large datasets. See the aforementioned references for further comments.

Choosing different options for the argument test simply manipulates the 'direction' of the p-values. That is, plotting tolerance contours at a significance level of 0.05 for a p-value surface calculated with test = "double" is equivalent to plotting tolerance contours at significance levels of 0.025 and 0.975 for test = "upper".

Implementation of the Monte-Carlo contours for the fixed-bandwidth estimator simply involves random allocation of case/control marks and re-estimation of the risk surface ITER times, against which the original estimate is compared. The bandwidth(s) for case and control densities in the permuted estimates are controlled by hpsim. If your risk surface is adaptive, hpsim is used to control the pilot bandwidths, h0sim the global bandwidths. In particular, for the adaptive symmetric estimator (Davies et al. 2015), it is assumed that the original estimate was itself calculated as a symmetric estimate via use of the pdef argument. The symchoice argument governs the specific permuted data set to use for the pilot estimate, and if hpsim is NULL, the pilot bandwidth thereof is taken from the stored pdef object in the original estimate. An error will occur if you attempt to set symchoice with an rs argument in this function that does not contain density estimates with present pdef objects of class "bivden". See the help file for bivariate.density for details on using the pdef argument.

In addition to the usage noted above, you may define either hpsim and/or h0sim as functions which re-calculate the case and control pilot (or fixed) bandwidth(s) and the global bandwidth(s) at each iteration, based on the data set of the permuted case/control marks. If so, these must strictly be functions that take the case data as the first argument and the control data as the second argument, each as a two-column matrix of the x-y coordinates. The function must strictly return a numeric vector of length 1 or 2; these entries to be assigned to the relevant density estimates as per the usage notes on supply of a numeric vector for hpsim. Take care – warnings will be issued if, for example, you specify a hpsim function that returns two numeric values, but your rs object is a symmetric-adaptive estimate (in which case it only makes sense to yield one pilot bandwidth)!

Value

A list with four components:

`X` the equally spaced sequence of length `ceiling(reduce*res)` giving the evaluation locations on the x axis (where `res` is the grid resolution as specified in the calls to `bivariate.density` for calculation of the densities for `rs` and `pooled`)

`Y` as above, for the y axis

`Z` a numeric `ceiling(reduce*res)*ceiling(reduce*res)` matrix giving the values of the risk surface over the evaluation grid. Values corresponding to grid coordinates outside the study region are assigned `NA`. If `method = "MC"`, this will be a single value of `NA`

`P` a `ceiling(reduce*res)*ceiling(reduce*res)` matrix giving the p-values corresponding to the evaluation grid in light of the elected `test`

Warning

Though far computationally intensive than calculation of Monte-Carlo p-value surfaces, the asymptotic p-value surfaces (particularly for adaptive relative risk surfaces) can still take some time to complete. The argument of `reduce` provides an option to reduce this computation time by decreasing the resolution of the evaluation grid. However, the accuracy and appearance of the resulting tolerance contours can be severely degraded if `reduce` is assigned too small a value. Care must therefore be taken and consideration given to the resolution of the original evaluation grid when altering `reduce` from its default value. For most practical purposes, we have found a value of `reduce` resulting in evaluation of a p-value surface of size 50 by 50 is adequate.

The MC approach is currently the only way to obtain tolerance contours for the adaptive-symmetric estimator. Given the computational expense, especially for adaptive estimates, it's recommended you do some preliminary runs with a small `ITER` value and/or make use of `reduce` to get acceptable running times.

Author(s)

T.M. Davies and M.L. Hazelton

References

Kelsall, J.E. and Diggle, P.J. (1995), Kernel estimation of relative risk, *Bernoulli*, **1**, 3-16.

Davies, T.M. and Hazelton, M.L. (2010), Adaptive kernel estimation of spatial relative risk, *Statistics in Medicine*, **29**(23) 2423-2437.

Davies, T.M., Jones, K. and Hazelton, M.L. (2015), Symmetric adaptive smoothing regimens for estimation of the spatial relative risk function, *Submitted for publication*.

Hazelton, M.L. and Davies, T.M. (2009), Inference based on kernel estimates of the relative risk function in geographical epidemiology, *Biometrical Journal*, **51**(1), 98-109.

Examples

```
## Not run:
data(chorley)
ch.h <- LSCV.density(chorley, hlim = c(0.1, 2))

ch.pool <- bivariate.density(data = chorley, pilotH = ch.h,
 adaptive = FALSE)
ch.case <- bivariate.density(data = chorley, ID = "larynx", pilotH = ch.h,
 adaptive = FALSE)
ch.con <- bivariate.density(data = chorley, ID = "lung", pilotH = ch.h,
 adaptive = FALSE)

##Compute log-risk surface and asymptotic p-value surfaces
ch.rrs <- risk(f = ch.case, g = ch.con)
ch.tol <- tolerance(rs = ch.rrs, pooled = ch.pool)
contour(ch.tol$X, ch.tol$Y, ch.tol$P, levels = 0.05, add = TRUE)



data(PBC)
PBC.casedata <- split(PBC)[[1]]
PBC.controldata <- split(PBC)[[2]]

pbc.rrs.rawdata <- risk(f = PBC.casedata, g = PBC.controldata,
 adaptive = TRUE, tolerate = TRUE)

plot(pbc.rrs.rawdata, display = "3d", aspect = 1:2, col = heat.colors(12)[12:1],
 tolerance.matrix = pbc.rrs.rawdata$P, tol.opt = list(col = "white", raise = 0.03))


## End(Not run)
```

Index

∗Topic **datasets**
    PBC, 20
∗Topic **package**
    sparr-package, 2

as.im, *5*
as.im.bivden, 5
as.im.rrs *(*as.im.bivden*)*, 5

bivariate.density, *3*, *5*, 6, *21–25*, *28–30*
bivden *(*bivariate.density*)*, 6
bw.relrisk, *14*, *17*

chorley, *2*
contour, *22*

data.frame, *6*, *8*, *13–19*, *23*
density.ppp, *8*

im, *5*

KBivN, *3*, 11
KBivQ, *3*, 12

list, *6*, *8*, *13–19*, *23*
LSCV.density, *3*, 13
LSCV.risk, *3*, 15

matrix, *6*, *8*, *13–19*, *23*

NS, *2*, 17

OS, *2*, 19, *24*
owin, *7*, *9*, *13*, *15*, *24*

par, *22*
par3d, *22*
PBC, *2*, 20
persp, *22*
persp3d, *22*
plot.bivden, *3*, 21
plot.default, *22*
plot.im, *22*
plot.rrs, *3*
plot.rrs *(*plot.bivden*)*, 21
ppp, *2*, *6–8*, *13–19*, *23, 24*
ppp.object, *8*, *13–16*, *18–20*
print.bivden, *3*
print.bivden *(*summary.bivden*)*, 26
print.rrs, *3*
print.rrs *(*summary.rrs*)*, 27

rgl, *3*, *22*
risk, *3*, *5*, *21, 22*, 23, *24, 25*, *27, 28*
rrs *(*risk*)*, 23

sparr, *3*, *7*, *20*
sparr *(*sparr-package*)*, 2
sparr-package, 2
spatstat, *2, 3*, *5*, *7, 8*
summary.bivden, *3*, 26
summary.rrs, *3*, 27

tolerance, *3*, *22*, *24, 25*, 27