Package ‘decon’

October 20, 2021

Type Package

Title Deconvolution Estimation in Measurement Error Models

Version 1.3-4

Date 2021-10-20

Author Xiao-Feng Wang, Bin Wang

Maintainer Xiao-Feng Wang <wangx6@ccf.org>

Description A collection of functions to deal
with nonparametric measurement error problems using
deconvolution kernel methods. We focus two measurement error
models in the package: (1) an additive measurement error model,
where the goal is to estimate the density or distribution
function from contaminated data; (2) nonparametric regression
model with errors-in-variables. The R functions allow the
measurement errors to be either homoscedastic or
heteroscedastic. To make the deconvolution estimators
computationally more efficient in R, we adapt the ``Fast Fourier
Transform'' (FFT) algorithm for density estimation with
error-free data to the deconvolution kernel estimation. Several
methods for the selection of the data-driven smoothing
parameter are also provided in the package. See details in:
Wang, X.F. and Wang, B. (2011). Deconvolution estimation in
measurement error models: The R package decon. Journal of
Statistical Software, 39(10), 1-24.

License GPL (>= 3)

Repository CRAN

Date/Publication 2021-10-20 20:50:05 UTC

NeedsCompilation yes

R topics documented:

`bw.dboot1` *A bootstrap bandwidth selection without resampling*

Description

To compute the optimal bandwidth using the bootstrap-type method without generation of any bootstrap sample.

Usage

```
bw.dboot1(y,sig, h0="dnrd", error="normal", grid=100, ub=2)
```

Arguments

`y`	The observed data. It is a vector of length at least 3.
`sig`	The standard deviation(s) σ. For homoscedastic errors, sig is a single value. Otherwise, sig is a vector of variances having the same length as y.
`h0`	An initial bandwidth parameter. The default vaule is the estimate from bw.dnrd.
`error`	Error distribution types: 'normal', 'laplacian' for normal and Laplacian errors, respectively.
`grid`	the grid number to search the optimal bandwidth when a bandwidth selector was specified in bw. Default value "grid=100".
`ub`	the upper boundary to search the optimal bandwidth, default value is "ub=2".

Details

Three cases are supported: (1) homo normal; (2) homo laplacian; (3) hetero normal.

Case (3) could be very slow, we reduce the number of grid points in computing the L-2 distance to 100 and reduce the optimal bandwidth searching grid points to 50 to speed up the algorithm.

The integration was approximated by computing the average over a fine grid of points (1000 points).

The case of heteroscedastic laplacian errors is not supported and is to be developed.

Value

the selected bandwidth.

Author(s)

X.F. Wang <wangx6@ccf.org>

B. Wang <bwang@jaguar1.usouthal.edu>

References

Delaigle, A. and Gijbels, I. (2004). Practical bandwidth selection in deconvolution kernel density estimation. *Computational Statistics and Data Analysis*, 45, 249-267.

Wang, X.F. and Wang, B. (2011). Deconvolution estimation in measurement error models: The R package decon. *Journal of Statistical Software*, 39(10), 1-24.

See Also

bw.dnrd, bw.dmise, bw.dboot2.

Examples

```
n <- 1000
x <- c(rnorm(n/2,-2,1),rnorm(n/2,2,1))
## the case of homoscedastic normal error
sig <- .8
u <- rnorm(n, sd=sig)
w <- x+u
bw.dboot1(w,sig=sig)
## the case of homoscedastic laplacian error
sig <- .8
## generate laplacian errors
u <- ifelse(runif(n) > 0.5, 1, -1) * rexp(n,rate=1/sig)
w <- x+u
bw.dboot1(w,sig=sig,error='laplacian')
## the case of heteroscedastic normal error
sig <- runif(n, .7, .9)
u <- sapply(sig, function(x) rnorm(1, sd=x))
w <- x+u
bw.dboot1(w,sig=sig,error='normal')
```

bw.dboot2 *A bootstrap bandwidth selection with resampling*

Description

To compute the optimal bandwidth using the bootstrap method with resampling.

Usage

```
bw.dboot2(y,sig,h0='dboot1',error='normal',B=1000,grid=100,ub=2)
```

Arguments

`y`	The observed data. It is a vector of length at least 3.
`sig`	The standard deviation(s) σ. For homoscedastic errors, sig is a single value. Otherwise, sig is a vector of variances having the same length as y.
`h0`	An initial bandwidth parameter. The default vaule is the estimate from bw.dboot1.
`error`	Error distribution types: 'normal', 'laplacian' for normal and Laplacian errors, respectively.
`B`	Bootstrap number, default value 1000.
`grid`	the grid number to search the optimal bandwidth when a bandwidth selector was specified in bw. Default value "grid=100".
`ub`	the upper boundary to search the optimal bandwidth, default value is "ub=2".

Details

Three cases are supported: (1) homo normal; (2) homo laplacian.

The integration was approximated by computing the average over a fine grid of points (1000 points).

Value

the selected bandwidth.

Author(s)

X.F. Wang <wangx6@ccf.org>

B. Wang <bwang@jaguar1.usouthal.edu>

References

Wang, X.F. and Wang, B. (2011). Deconvolution estimation in measurement error models: The R package decon. *Journal of Statistical Software*, 39(10), 1-24.

See Also

`bw.dnrd`, `bw.dmise`, `bw.dboot1`.

Examples

```
n <- 1000
x <- c(rnorm(n/2,-2,1),rnorm(n/2,2,1))
## the case of homoscedastic normal error
sig <- .8
u <- rnorm(n, sd=sig)
w <- x+u
bw.dboot2(w,sig=sig)
## the case of homoscedastic laplacian error
sig <- .8
## generate laplacian error
u <- ifelse(runif(n) > 0.5, 1, -1) * rexp(n,rate=1/sig)
w <- x+u
bw.dboot2(w,sig=sig,error='laplacian')
```

---

bw.dmise — *The MISE based plug-in bandwidth selection*

Description

To compute the optimal bandwidth using the plug-in methods by minimizing MISE.

Usage

```
bw.dmise(y,sig,error="normal",kernel="support",grid=100,ub=2)
```

Arguments

`y`	The observed data. It is a vector of length at least 3.
`sig`	The standard deviation(s) σ. For homoscedastic errors, sig is a single value. Otherwise, sig is a vector of variances having the same length as y.
`error`	Error distribution types: 'normal', 'laplacian' for normal and Laplacian errors, respectively.
`kernel`	Kernel type: 'support' for support kernel; and 'normal' for Gaussian kernel.
`grid`	the grid number to search the optimal bandwidth when a bandwidth selector was specified in bw. Default value "grid=100".
`ub`	the upper boundary to search the optimal bandwidth, default value is "ub=2".

Details

The current version approximate the second term in the MISE by assuming that X is normally distributed.

Value

the selected bandwidth.

Author(s)

X.F. Wang <wangx6@ccf.org>

B. Wang <bwang@jaguar1.usouthal.edu>

References

Fan, J. (1992). Deconvolution with supersmooth distributions. *The Canadian Journal of Statistics*, 20, 155-169.

Stefanski, L. and Carroll, R. J. (1990). Deconvoluting kernel density estimators. *Statistics*, 21, 169-184.

Wang, X.F. and Wang, B. (2011). Deconvolution estimation in measurement error models: The R package decon. *Journal of Statistical Software*, 39(10), 1-24.

See Also

bw.dnrd, bw.dboot1, bw.dboot2.

Examples

```
n <- 1000
x <- c(rnorm(n/2,-2,1),rnorm(n/2,2,1))
## the case of homoscedastic normal error
sig <- .8
u <- rnorm(n, sd=sig)
w <- x+u
bw.dmise(w,sig=sig,error='normal');
## The small error case
sig <- .25
u <- rnorm(n, sd=sig)
w <- x+u
bw.dmise(w,sig=sig,kernel='normal',error='normal');

## the case of homoscedastic laplacian error
sig <- .8
## generate laplacian error
u <- ifelse(runif(n) > 0.5, 1, -1) * rexp(n,rate=1/sig)
w <- x+u
bw.dmise(w,sig=sig,error='laplace')

## the case of heteroscedastic normal error
sig <- runif(n, .7, .9)
u <- sapply(sig, function(x) rnorm(1, sd=x))
w <- x+u
bw.dmise(w,sig=sig,kernel='support',error='normal')
```

bw.dnrd *A rule of thumb bandwidth selection in denconvolution problems*

Description

To compute the optimal bandwidth using the rule-of-thumb methods based on theorem 1 and theorem 2 of Fan (1991).

Usage

```
bw.dnrd(y,sig,error='normal')
```

Arguments

y	The observed data. It is a vector of length at least 3.
sig	The standard deviation(s) σ. For homoscedastic errors, sig is a single value. otherwise, sig is a vector of variances having the same length as y.
error	Error distribution types: 'normal', 'laplacian' for normal and Laplacian errors, respectively.

Details

The current version approximate the second term in the MISE by assuming that X is normally distributed. In the case of heteroscedastic error, the variance was approximated by the arithematic mean of the variances of U.

Value

the selected bandwidth.

Author(s)

X.F. Wang <wangx6@ccf.org>

B. Wang <bwang@jaguar1.usouthal.edu>

References

Fan, J. (1991). On the optimal rates of convergence for nonparametric deconvolution problems. *The Annals of Statistics*, 19, 1257-1272.

Fan, J. (1992). Deconvolution with supersmooth distributions. *The Canadian Journal of Statistics*, 20, 155-169.

Stefanski, L. and Carroll, R. J. (1990). Deconvoluting kernel density estimators. *Statistics*, 21, 169-184.

Wang, X.F. and Wang, B. (2011). Deconvolution estimation in measurement error models: The R package decon. *Journal of Statistical Software*, 39(10), 1-24.

See Also

bw.dmise, bw.dboot1, bw.dboot2.

Examples

```
n <- 1000
x <- c(rnorm(n/2,-2,1),rnorm(n/2,2,1))
## the case of homoscedastic normal error
sig <- .8
u <- rnorm(n, sd=sig)
w <- x+u
bw.dnrd(w,sig=sig)
## the case of homoscedastic laplacian error
sig <- .8
## generate laplacian errors
u <- ifelse(runif(n) > 0.5, 1, -1) * rexp(n,rate=1/sig)
w <- x+u
bw.dnrd(w,sig=sig,error='laplacian')
## the case of heteroscedastic normal error
sig <- runif(n, .7, .9)
u <- sapply(sig, function(x) rnorm(1, sd=x))
w <- x+u
bw.dnrd(w,sig=sig,error='normal')
```

DeconCdf — *Estimating cumulative distribution function from data with measurement error*

Description

To compute the cumulative distribution function from data coupled with measurement error. The measurement errors can be either homoscedastic or heteroscedastic.

Usage

```
DeconCdf(y,sig,x,error="normal",bw="dboot1",adjust=1,
n=512,from,to,cut=3,na.rm=FALSE,grid=100,ub=2,...)
```

Arguments

y	The observed data. It is a vector of length at least 3.
sig	The standard deviations σ. If homoscedastic errors, sig is a single value. If heteroscedastic errors, sig is a vector of standard deviations having the same length as y.
x	x is user-defined grids where the CDF will be evaluated. FFT method is not applicable if x is given.
error	Error distribution types: (1) 'normal' for normal errors; (2) 'laplacian' for Laplacian errors; (3) 'snormal' for a special case of small normal errors.

bw	Specifies the bandwidth. It can be a single numeric value which has been predetermined; or computed with the specific bandwidth selector: 'dnrd' to compute the rule-of-thumb plugin bandwidth as suggested by Fan (1991); 'dmise' to compute the plugin bandwidth by minimizing MISE; 'dboot1' to compute the bootstrap bandwidth selector without resampling (Delaigle and Gijbels, 2004a), which minimizing the MISE bootstrap bandwidth selectors; 'boot2' to compute the smoothed bootstrap bandwidth selector with resampling.
adjust	adjust the range there the CDF is to be evaluated. By default, $adjust = 1$.
n	number of points where the CDF is to be evaluated.
from	the starting point where the CDF is to be evaluated.
to	the starting point where the CDF is to be evaluated.
cut	used to adjust the starting end ending points where the CDF is to be evaluated.
na.rm	is set to FALSE by default: no NA value is allowed.
grid	the grid number to search the optimal bandwidth when a bandwidth selector was specified in bw. Default value "grid=100".
ub	the upper boundary to search the optimal bandwidth, default value is "ub=2".
...	control

Details

FFT is currently not supported for CDF computing.

Value

An object of class "Decon".

Author(s)

X.F. Wang <wangx6@ccf.org>

B. Wang <bwang@jaguar1.usouthal.edu>

References

Delaigle, A. and Gijbels, I. (2004). Bootstrap bandwidth selection in kernel density estimation from a contaminated sample. *Annals of the Institute of Statistical Mathematics*, 56(1), 19-47.

Fan, J. (1991). On the optimal rates of convergence for nonparametric deconvolution problems. *The Annals of Statistics*, 19, 1257-1272.

Fan, J. (1992). Deconvolution with supersmooth distributions. *The Canadian Journal of Statistics*, 20, 155-169.

Hall, P. and Lahiri, S.N. (2008). Estimation of distributions, moments and quantiles in deconvolution problems. *Annals of Statistics*, 36(5), 2110-2134.

Stefanski L.A. and Carroll R.J. (1990). Deconvoluting kernel density estimators. *Statistics*, 21, 169-184.

Wang, X.F., Fan, Z. and Wang, B. (2010). Estimating smooth distribution function in the presence of heterogeneous measurement errors. *Computational Statistics and Data Analysis*, 54, 25-36.

Wang, X.F. and Wang, B. (2011). Deconvolution estimation in measurement error models: The R package decon. *Journal of Statistical Software*, 39(10), 1-24.

See Also

DeconPdf, DeconNpr.

Examples

```
#####################
## the R function to estimate the smooth distribution function
#SDF <- function (x, bw = bw.nrd0(x), n = 512, lim=1){
#        dx <- lim*sd(x)/20
#        xgrid <- seq(min(x)-dx, max(x)+dx, length = n)
#        Fhat <- sapply(x, function(x) pnorm((xgrid-x)/bw))
#        return(list(x = xgrid, y = rowMeans(Fhat)))
#    }

## Case study: homoscedastic normal errors
n2 <- 100
x2 <- c(rnorm(n2/2,-3,1),rnorm(n2/2,3,1))
sig2 <- .8
u2 <- rnorm(n2, sd=sig2)
w2 <- x2+u2
# estimate the bandwidth with the bootstrap method with resampling
bw2 <- bw.dboot2(w2,sig=sig2, error="normal")
# estimate the distribution function with measurement error
F2 <-  DeconCdf(w2,sig2,error='normal',bw=bw2)
plot(F2,  col="red", lwd=3, lty=2, xlab="x", ylab="F(x)", main="")

#lines(SDF(x2), lwd=3, lty=1)
#lines(SDF(w2), col="blue", lwd=3, lty=3)
```

DeconCPdf — *Estimating conditional probability density function from data with measurement error*

Description

To compute the conditional probability density function from data with measurement error. The measurement errors have to be homoscedastic.

Usage

```
DeconCPdf(y,sig,y0,error='normal',bw1='dboot1',bw2='nrd0',adjust=1,
fft=FALSE,n=512,from,to,cut=3,na.rm=FALSE,grid=100,ub=2,tol=0,...)
```

Arguments

y	The observed data. It is a vector of length at least 3.
sig	The standard deviations σ. If homoscedastic errors, sig is a single value. If heteroscedastic errors, sig is a vector of standard deviations having the same length as y.
y0	The given conditional data point in the conditional density f(xly=y0).
error	Error distribution types: (1) 'normal' for normal errors; (2) 'laplacian' for Laplacian errors; (3) 'snormal' for a special case of small normal errors.
bw1	The bandwidth for the deconvolution density f_X. It can be a single numeric value which has been pre-determined; or computed with the specific bandwidth selector: 'dnrd' to compute the rule-of-thumb plugin bandwidth as suggested by Fan (1991); 'dmise' to compute the plugin bandwidth by minimizing MISE; 'dboot1' to compute the bootstrap bandwidth selector without resampling (Delaigle and Gijbels, 2004a), which minimizing the MISE bootstrap bandwidth selectors; 'boot2' to compute the smoothed bootstrap bandwidth selector with resampling.
bw2	The bandwidth for the kernel density f_Y. It can be a single numeric value which has been pre-determined; or computed with the specific bandwidth selector: 'nrd0','nrd','ucv', 'bcv', and 'SJ' (see the "density" function in R).
adjust	adjust the range there the PDF is to be evaluated. By default, $adjust = 1$.
fft	To specify the method to compute the PDF. 'fft=FALSE' to compute directly; 'fft=TRUE' to compute the PDF by using the Fast Fourier Transformation.
n	number of points where the conditional PDF is to be evaluated.
from	the starting point where the conditional PDF is to be evaluated.
to	the starting point where the conditional PDF is to be evaluated.
cut	used to adjust the starting end ending points where the conditional PDF is to be evaluated.
na.rm	is set to FALSE by default: no NA value is allowed.
grid	the grid number to search the optimal bandwidth when a bandwidth selector was specified in bw. Default value "grid=100".
ub	the upper boundary to search the optimal bandwidth, default value is "ub=2".
tol	the parameter to avoid the estimate of f(ylx) too small. The default vaule is 0. It can not exceed 0.05.
...	control

Details

If the number of points to be evaluated is too small (less than 32), a direct computing method is preferred. The current version can support up to 2^21 points where the conditional PDF to be computed.

Value

An object of class "Decon".

Author(s)

X.F. Wang <wangx6@ccf.org>

B. Wang <bwang@jaguar1.usouthal.edu>

References

Fan, J. (1991). On the optimal rates of convergence for nonparametric deconvolution problems. *The Annals of Statistics*, 19, 1257-1272.

Wang XF, Ye D (2010). Conditional density estimation with measurement error. Technical Report.

Wang, X.F. and Wang, B. (2011). Deconvolution estimation in measurement error models: The R package decon. *Journal of Statistical Software*, 39(10), 1-24.

See Also

DeconPdf.

Examples

```
n <- 100
x <- c(rnorm(n/2,-2,1),rnorm(n/2,2,1))
sig <- .8
u <- rnorm(n,sd=sig)
w <- x+u

f1 <-  DeconCPdf(w,sig, y0=-2.5, error='normal')
#f2 <-  DeconCPdf(w,sig, y0=0, error='normal')
#f3 <-  DeconCPdf(w,sig, y0=2.5, error='normal')

#par(mfrow=c(2,2))
plot(density(w), main="f_w", xlab="w")
plot(f1, main="f1", xlab="x")
#plot(f2, main="f2", xlab="x")
#plot(f3, main="f3", xlab="x")
```

DeconNpr — *Perform nonparametric regression with errors-in-variables*

Description

To perform nonparametric regression with errors-in-variables. The measurement errors can be either homoscedastic or heteroscedastic.

Usage

```
DeconNpr(y,sig,z,x,error='normal',bw='dboot1',adjust=1,
n=512,from,to,cut=3,na.rm=FALSE,grid=100,ub=2,...)
```

Arguments

y	The observed data. It is a vector of length at least 3.
sig	The standard deviations σ. If homoscedastic errors, sig is a single value. If heteroscedastic errors, sig is a vector of standard deviations having the same length as y.
z	z is the dependent variable.
x	x is user-defined grids where the regression function will be evaluated. FFT method is not applicable if x is given.
error	Error distribution types: (1) 'normal' for normal errors; (2) 'laplacian' for Laplacian errors; (3) 'snormal' for a special case of small normal errors.
bw	Specifies the bandwidth. It can be a single numeric value which has been predetermined; or computed with the specific bandwidth selector: 'dnrd' to compute the rule-of-thumb plugin bandwidth as suggested by Fan (1991); 'dmise' to compute the plugin bandwidth by minimizing MISE; 'dboot1' to compute the bootstrap bandwidth selector without resampling (Delaigle and Gijbels, 2004a), which minimizing the MISE bootstrap bandwidth selectors; 'boot2' to compute the smoothed bootstrap bandwidth selector with resampling.
adjust	adjust the range there the regression function is to be evaluated. By default, $adjust = 1$.
n	number of points where the regression function is to be evaluated.
from	the starting point where the regression functionF is to be evaluated.
to	the starting point where the regression function is to be evaluated.
cut	used to adjust the starting end ending points where the regression function is to be evaluated.
na.rm	is set to FALSE by default: no NA value is allowed.
grid	the grid number to search the optimal bandwidth when a bandwidth selector was specified in bw. Default value "grid=100".
ub	the upper boundary to search the optimal bandwidth, default value is "ub=2".
...	control

Details

FFT is currently not supported for nonparametric regression.

Value

An object of class "Decon".

Author(s)

X.F. Wang <wangx6@ccf.org>

B. Wang <bwang@jaguar1.usouthal.edu>

References

Fan, J. and Truong, Y.K. (1993). Nonparametric regression with errors in variables. *Annals of Statistics*, 21(4), 1900-1925.

Delaigle, A. and Meister, A. (2007). Nonparametric regression estimation in the heteroscedastic errors-in-variables problem. *Journal of the American Statistical Association*, 102, 1416-1426.

Wang, X.F. and Wang, B. (2011). Deconvolution estimation in measurement error models: The R package decon. *Journal of Statistical Software*, 39(10), 1-24.

See Also

DeconPdf, DeconCdf.

Examples

```
n <- 2000
x <- c(rnorm(n/2,2,1), rnorm(n/2, -2,1))
sig <- .8
u <- sig*rnorm(n)
w <- x+u
e <- rnorm(n, sd=0.2)
y <- x^2-2*x+e
bw1 <- bw.dboot1(w, sig)
# estimate the unknown density with measurement error
(m1 <-  DeconNpr(w, sig, y ,error="normal", from=0.9*min(x), to=0.9*max(x)))
# plot the results
plot(m1, col="red", lwd=3, lty=2, xlab="x", ylab="m(x)", main="",
zero.line=FALSE)
lines(ksmooth(x,y, kernel = "normal", 2, range.x=c(0.9*min(x),0.9*max(x))),
lwd=3, lty=1)
lines(ksmooth(w,y, kernel = "normal", 2, range.x=c(0.9*min(x),0.9*max(x))),
col="blue", lwd=3, lty=3)
```

DeconPdf *Estimating probability density function from data with measurement error*

Description

To compute the probability density function from data contaminated with measurement error. The measurement errors can be either homoscedastic or heteroscedastic.

Usage

```
DeconPdf(y,sig,x,error='normal',bw='dboot1',adjust=1,fft=FALSE,
n=512,from,to,cut=3,na.rm=FALSE,grid=100,ub=2, ...)
```

Arguments

y	The observed data. It is a vector of length at least 3.
sig	The standard deviations σ. If homoscedastic errors, sig is a single value. If heteroscedastic errors, sig is a vector of standard deviations having the same length as y.
x	x is user-defined grids where the PDF will be evaluated. FFT method is not applicable if x is given.
error	Error distribution types: (1) 'normal' for normal errors; (2) 'laplacian' for Laplacian errors; (3) 'snormal' for a special case of small normal errors.
bw	Specifies the bandwidth. It can be a single numeric value which has been predetermined; or computed with the specific bandwidth selector: 'dnrd' to compute the rule-of-thumb plugin bandwidth as suggested by Fan (1991); 'dmise' to compute the plugin bandwidth by minimizing MISE; 'dboot1' to compute the bootstrap bandwidth selector without resampling (Delaigle and Gijbels, 2004a), which minimizing the MISE bootstrap bandwidth selectors; 'boot2' to compute the smoothed bootstrap bandwidth selector with resampling.
adjust	adjust the range there the PDF is to be evaluated. By default, $adjust = 1$.
fft	To specify the method to compute the PDF. 'fft=FALSE' to compute directly; 'fft=TRUE' to compute the PDF by using the Fast Fourier Transformation.
n	number of points where the PDF is to be evaluated.
from	the starting point where the PDF is to be evaluated.
to	the starting point where the PDF is to be evaluated.
cut	used to adjust the starting end ending points where the PDF is to be evaluated.
na.rm	is set to FALSE by default: no NA value is allowed.
grid	the grid number to search the optimal bandwidth when a bandwidth selector was specified in bw. Default value "grid=100".
ub	the upper boundary to search the optimal bandwidth, default value is "ub=2".
...	control

Details

If the number of points to be evaluated is too small (less than 32), a direct computing method is preferred. The current version can support up to $2^2 1$ points where the PDF to be computed.

Value

An object of class "Decon".

Author(s)

X.F. Wang <wangx6@ccf.org>

B. Wang <bwang@jaguar1.usouthal.edu>

References

Delaigle, A. and Meister, A. (2008). Density estimation with heteroscedastic error. *Bernoulli*, 14, 562-579.

Fan, J. (1991). On the optimal rates of convergence for nonparametric deconvolution problems. *The Annals of Statistics*, 19, 1257-1272.

Fan, J. (1992). Deconvolution with supersmooth distributions. *The Canadian Journal of Statistics*, 20, 155-169.

Wang, X.F. and Wang, B. (2011). Deconvolution estimation in measurement error models: The R package decon. *Journal of Statistical Software*, 39(10), 1-24.

See Also

DeconCdf, DeconNpr, DeconCPdf.

Examples

```
## Deconvolution: the case of homoscedastic errors
## Case 1.1: homoscedastic Laplacian errors
n1 <- 500
x1 <- rnorm(n1, sd=1)
sig1 <- .5
u1 <- ifelse(runif(n1) > 0.5, 1, -1) * rexp(n1,rate=1/sig1)
w1 <- x1+u1
## The rule-of-thumb method may not be accurate,
## you may try the bootstrap method
bw1 <- bw.dnrd(w1,sig=sig1, error="laplacian")
f1 <-  DeconPdf(w1,sig1,error='laplacian',bw=bw1, fft=TRUE)

## Case 1.2: homoscedastic normal errors
#n2 <- 1000
#x2 <- c(rnorm(n2/2,-3,1),rnorm(n2/2,3,1))
#sig2 <- .8
#u2 <- rnorm(n2, sd=sig2)
#w2 <- x2+u2
# estimate the bandwidth with the bootstrap method with resampling
#bw2 <- bw.dboot2(w2,sig=sig2, error="normal")
# estimate the unknown density with measurement error
#(f2 <-  DeconPdf(w2,sig2,error='normal',bw=bw2, fft=TRUE))

# plot the results
#par(mfrow=c(1,2))
#plot(f1,  col="red", lwd=3, lty=2, xlab="x", ylab="f(x)", main="")
#lines(density(x1, from=min(w1), to=max(w1)), lwd=3, lty=1)
#lines(density(w1), col="blue", lwd=3, lty=3)
#plot(f2,  col="red", lwd=3, lty=2, xlab="x", ylab="f(x)", main="")
#lines(density(x2, from=min(w2), to=max(w2)), lwd=3, lty=1)
#lines(density(w2), col="blue", lwd=3, lty=3)
```

```
## Deconvolution: the case of heteroscedastic errors
## Case 2: heteroscedastic normal errors
#n3 <- 2000
#x3 <- rchisq(n3, df=1.5, ncp=0)
#sig3 <- 0.7+ x3/max(x3)
#u3 <- sapply(sig3, function(x) rnorm(1, sd=x))
#w3 <- x3+u3
# estimate the bandwidth using the bootstrap method withou resampling
#bw3 <- bw.dboot1(w3,sig=sig3, error="normal")
# estimate the unknown density with measurement error
#(f3 <-  DeconPdf(w3,sig3,error="normal", bw=bw3, fft=TRUE))

# plot the results
#par(mfrow=c(1,1))
#plot(f3,  col="red", lwd=3, lty=2, ylim=c(0,0.4), xlab="x", ylab="f(x)", main="")
#lines(density(x3, adjust=2), lwd=3, lty=1)
#lines(density(w3, adjust=2), col="blue", lwd=3, lty=3)
```

---

framingham *Framingham Data*

Description

The data is from the Framingham Study on coronary heart disease described by Carroll et al. (2006). The data consist of measurements of systolic blood pressure (SBP) obtained at two different examinations in 1,615 males on an 8-year follow-up from the first examination. At each examination, the SBP was measured twice and for each individual.

Usage

```
data(framingham)
```

Format

A data frame with 1615 observations on 4 variables.

SBP11	numeric	First systolic blood pressure at exam 1
SBP12	numeric	Second systolic blood pressure at exam 1
SBP21	numeric	First systolic blood pressure at exam 2
SBP22	numeric	Second systolic blood pressure at exam 2

References

Carroll, R.~J., Ruppert, D., Stefanski, L.~A., and Crainiceanu, C. (2006). *Measurement Error in Nonlinear Models: A Modern Perspective, Second Edition*. Chapman Hall, New York.

See Also

DeconPdf

---

galaxy — *The observed position-velocity data of low surface brightness galaxies*

Description

The astronomical position-velocity data is from a sample of 26 low surfaces brightness (LSB) galaxies. The data contain 318 stars with their radiuses in kiloparsec (kpc), observed velocities of stars in km/s (relative to center, corrected for inclination) from 26 LSB galaxies. It was known that the velocities were measured with errors. In the data set, each velocity includes its estimated standard deviation of measurement errors.

Usage

```
data(galaxy)
```

Format

A data frame with 318 observations on 3 variables.

Rkpc	numeric	radiuses in kiloparsec (kpc)
V	numeric	velocities of stars in km/s
Err	numeric	standard deviations of measurement errors in measuring velocities

References

De Blok, W.J.G., McGaugh, S.S., and Rubin, V.C. (2001). High-resolution rotation curves of low surface brightness galaxies: Mass Models. *The Astronomical Journal*, 122, 2396-2427.

See Also

DeconNpr

---

npdenest — *Estimating probability density function from data with unknown measurement error*

Description

To compute the probability density function from data contaminated with measurement error. The measurement error type is unknown.

Usage

```
npdenest(w, e, bw, adjust = 1, n = 512, from, to, cut = 3, na.rm = FALSE,...)
```

Arguments

w	The observed data. It is a vector of length at least 3.
e	Observed vector of measurement errors.
bw	Smoothing parameter.
adjust	adjust the range there the PDF is to be evaluated. By default, $adjust = 1$.
n	number of points where the PDF is to be evaluated.
from	the starting point where the PDF is to be evaluated.
to	the starting point where the PDF is to be evaluated.
cut	used to adjust the starting end ending points where the PDF is to be evaluated.
na.rm	is set to FALSE by default: no NA value is allowed.
...	controls

Details

The optimal bandwidth is selected by minimizing `abs(Var(f.hat)+Var(E)-Var(Y))`.

Value

An object of class "Decon".

References

Wang, X.F. and Wang, B. (2011). Deconvolution estimation in measurement error models: The R package decon. *Journal of Statistical Software*, 39(10), 1-24.

See Also

DeconCdf, DeconNpr, DeconCPdf.

Examples

```
n1 <- 1500
x1 <- rnorm(n1, sd=1)
sig1 <- .5
u1 <- ifelse(runif(n1) > 0.5, 1, -1) * rexp(n1,rate=1/sig1)
w1 <- x1+u1
## The rule-of-thumb method may not be accurate,
## you may try the bootstrap method
bw1 <- bw.dnrd(w1,sig=sig1, error="laplacian")
(f1 <-  DeconPdf(w1,sig1,error='laplacian',bw=bw1, fft=TRUE))
```

```
(f2 <-  npdenest(w1, u1))

# plot the results
par(mfrow=c(1,1))
plot(f1,  col="red", lwd=3, lty=2, xlab="x", ylab="f(x)", main="")
lines(density(x1, from=min(w1), to=max(w1)), lwd=3, lty=1)
lines(density(w1), col="blue", lwd=3, lty=3)
lines(f2, col='red', lty=1,lwd=3)
```

---

npreg   *Nonparametric regression based on data with unknown measurement error*

Description

To compute nonparametric regression line from data contaminated with measurement error. The measurement error type is unknown.

Usage

```
npreg(w, y, e, bw, adjust = 1, n = 512, from, to, cut = 0, na.rm = FALSE,...)
```

Arguments

w,y	The observed data W and Y. It is a vector of length at least 3.
e	Observed vector of measurement errors.
bw	Smoothing parameter.
adjust	adjust the range there the PDF is to be evaluated. By default, $adjust = 1$.
n	number of points where the PDF is to be evaluated.
from	the starting point where the PDF is to be evaluated.
to	the starting point where the PDF is to be evaluated.
cut	used to adjust the starting end ending points where the PDF is to be evaluated.
na.rm	is set to FALSE by default: no NA value is allowed.
...	controls

Details

The optimal bandwidth is selected by minimizing `abs(Var(f.hat)+Var(E)-Var(Y))`.

Value

An object of class “Decon”.

References

Wang, X.F. and Wang, B. (2011). Deconvolution estimation in measurement error models: The R package decon. *Journal of Statistical Software*, 39(10), 1-24.

See Also

DeconCdf, DeconNpr, DeconCPdf.

Examples

```
n <- 2000
x <- c(rnorm(n/2,2,1), rnorm(n/2, -2,1))
sig <- .8
u <- sig*rnorm(n)
w <- x+u
e <- rnorm(n, sd=0.2)
y <- x^2-2*x+e
bw1 <- bw.dboot1(w, sig)
u0 <- sig*rnorm(n/2) # typically the size of u0 is smaller than x.
m2 <- npreg(w, y, u0, from=0.9*min(x), to=0.9*max(x))

# plot the results
plot(m2, col="red", lwd=3, lty=2, xlab="x", ylab="m(x)", main="",
zero.line=FALSE)
lines(ksmooth(x,y, kernel = "normal", 2, range.x=c(0.9*min(x),0.9*max(x))),
lwd=3, lty=1)
lines(ksmooth(w,y, kernel = "normal", 2, range.x=c(0.9*min(x),0.9*max(x))),
col="blue", lwd=3, lty=3)
```

---

plot.Decon *Plot a Decon Object*

Description

To plot a “Decon” object generated by DeconPdf(...), DeconCdf(...), DeconCPdf(...), or DeconNpr(...).

Details

This function is to plot the estimated function generated by DeconPdf(...), DeconCdf(...), DeconCPdf(...), or DeconNpr(...)

Author(s)

X.F. Wang <wangx6@ccf.org>

B. Wang <bwang@jaguar1.usouthal.edu>

See Also

DeconPdf,DeconCdf,DeconCPdf,DeconNpr.

---

print.Decon *Print a Decon Object*

Description

To print a "Decon" object generated by DeconPdf(...), DeconCdf(...), DeconCPdf(...), or DeconNpr(...).

Details

This function is to print the summary description from the object generated by DeconPdf(...), DeconCdf(...), DeconCPdf(...), or DeconNpr(...)

Author(s)

X.F. Wang <wangx6@ccf.org>

B. Wang <bwang@jaguar1.usouthal.edu>

See Also

DeconPdf,DeconCdf,DeconCPdf,DeconNpr.

Index