Package 'sharpPen'

March 6, 2021

Version 1.7

Title Penalized Data Sharpening for Local Polynomial Regression

Author W.J. Braun [aut],
D. Wang [aut, cre],
X.J. Hu [aut, ctb]

Maintainer D. Wang <wdy@mail.ubc.ca>

Depends KernSmooth, MASS, glmnet, np, Matrix

Description Functions and data sets for penalized data sharpening.
Nonparametric regressions are computed subject to smoothness
and other kinds of penalties.

NeedsCompilation yes

Repository CRAN

Date/Publication 2021-03-06 05:40:03 UTC

R topics documented:

data_sharpening — *Penalized data sharpening for Local Linear, Quadratic and Cubic Regression*

Description

Penalized data sharpening has been proposed as a way to enforce certain constraints on a local polynomial regression estimator. In addition to a bandwidth, a coefficient of the penalty term is also required. We propose propose systematic approaches for choosing these tuning parameters, in part, by considering the optimization problem from the perspective of ridge regression.

Usage

```
data_sharpening(xx,yy,zz,p,h1=NULL,gammaest=NULL,penalty,lambda=NULL)
```

Arguments

`xx`	numeric vector of x data. Missing values are not accepted.
`yy`	numeric vector of y data. This must be same length as `x`, and missing values are not accepted.
`zz`	numeric vector of gridpoint z data. Missing values are not accepted.
`p`	degree of local polynomial used.
`h1`	the kernel bandwidth smoothing parameter. If NULL, this value will be estimated by function dpill for Local Linear Regression, and will be estimated by function dpilc for Local Quadratic and Cubic Regression.
`gammaest`	the shape constraint parameter. Cannot be NULL for Periodic shape constraint. Can be NULL for Exponential shape constraint.
`penalty`	the type of shape constraint, can be "Exponential" and "Periodicity".
`lambda`	a coefficient of the penalty term, default is NULL.

Value

the sharpened response variable.

Author(s)

D.Wang and W.J.Braun

Examples

```
set.seed(1234567)
gam<-4
gamest<-gam
g <- function(x) 3*sin(x*(gam*pi))+5*cos(x*(gam*pi))+6*x
sigma<-3
xx<-seq(0,1,length=100)
```

```
yy<-g(xx)+rnorm(100,sd=sigma)
zz<-xx
h1<-dpilc(xx,yy)
local_fit<-t(getA(h=h1,xx=xx,zz=zz,p=2))%*%yy
y_sharp<-data_sharpening(xx=xx,yy=yy,zz=zz,p=2,gammaest=gamest,penalty="Periodicity")
sharp_fit<-t(getA(h=h1,xx=xx,zz=zz,p=2))%*%y_sharp
plot(c(min(xx),max(xx)),c(min(yy)-0.5,max(yy)+0.5),type="n",,xlab="x",ylab="y")
legend("bottomright",legend=c("curve_local","curve_sharpen"), col=c(1,3),bty="n",pch=c("-","-"))
lines(xx,local_fit)
lines(xx,sharp_fit,col=3, lwd=2)
points(xx,yy,col= rgb(0.8,0.2,0.2,0.2))
```

`dpilc` *Select a Bandwidth for Local Quadratic and Cubic Regression*

Description

Use direct plug-in methodology to select the bandwidth of a local quadratic and local cubic Gaussian kernel regression estimate, as an extension of Wand's `dpill` function.

Usage

```
dpilc(xx, yy, blockmax = 5, divisor = 20, trim = 0.01,
                 proptrun = 0.05, gridsize = 401L, range.x = range(x))
```

Arguments

`xx`	numeric vector of x data. Missing values are not accepted.
`yy`	numeric vector of y data. This must be same length as `x`, and missing values are not accepted.
`blockmax`	the maximum number of blocks of the data for construction of an initial parametric estimate.
`divisor`	the value that the sample size is divided by to determine a lower limit on the number of blocks of the data for construction of an initial parametric estimate.
`trim`	the proportion of the sample trimmed from each end in the `x` direction before application of the plug-in methodology.
`proptrun`	the proportion of the range of `x` at each end truncated in the functional estimates.
`gridsize`	number of equally-spaced grid points over which the function is to be estimated.
`range.x`	vector containing the minimum and maximum values of `x` at which to compute the estimate. For density estimation the default is the minimum and maximum data values with 5% of the range added to each end. For regression estimation the default is the minimum and maximum data values.

Details

This function is a local cubic (also quadratic) extension of the `dpill` function of Wand's KernSmooth package. The kernel is the standard normal density. Least squares octic fits over blocks of data are used to obtain an initial estimate. As in Wand's implementation of the Ruppert, Sheather and Wand selector, Mallow's C_p is used to select the number of blocks. An option is available to make use of a periodic penalty (with possible trend) relating the 4th derivative of the regression function to a constant (gamma) times the 2nd derivative. This avoids the need to calculate the octic fits and reverts back to the original quartic fits of `dpill` with appropriate adjustments to the estimated functionals needed in the direct-plug-in bandwidth calculation. This code is similar to `dpilq` but uses a 6th degree polyomial approximation instead of an 8th degree polynomial approximation.

Value

the selected bandwidth.

Warning

If there are severe irregularities (i.e. outliers, sparse regions) in the `x` values then the local polynomial smooths required for the bandwidth selection algorithm may become degenerate and the function will crash. Outliers in the `y` direction may lead to deterioration of the quality of the selected bandwidth.

Author(s)

D.Wang and W.J.Braun

References

Ruppert, D., Sheather, S. J. and Wand, M. P. (1995). An effective bandwidth selector for local least squares regression. *Journal of the American Statistical Association*, **90**, 1257–1270.

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

See Also

`ksmooth`, `locpoly`.

Examples

```
x <- faithful$eruptions
y <- faithful$waiting
plot(x, y)
h <- dpill(x, y)
fit <- locpoly(x, y, bandwidth = h, degree=1)
lines(fit)
h <- dpilc(x, y)
fit <- locpoly(x, y, bandwidth = h, degree=2)
lines(fit, col=3, lwd=2)
fit <- locpoly(x, y, bandwidth = h, degree=3)
lines(fit, col=2, lwd=2)
```

getA *Local Polynomial Estimator Matrix Construction*

Description

Construct a matrix based on the local polynomial estimation at a corresponding sequence of x data and sequence of gridpoint z.

Usage

```
getA(h,xx,zz,p)
```

Arguments

h	the kernel bandwidth smoothing parameter.
xx	numeric vector of x data. Missing values are not accepted.
zz	numeric vector of gridpoint z data. Missing values are not accepted.
p	degree of local polynomial used.

Value

local polynomial estimator matrix

Author(s)

X.J. Hu

getB *Shape Constraint Matrix Construction*

Description

Construct a shape constraint matrix at a corresponding sequence of x data and sequence of gridpoint z.

Usage

```
getB(penalty,gamma,h, xx,zz,p)
```

Arguments

- `penalty` the type of shape constraint, can be "R1", "Roughness", "Exponential" and "Periodicity".
- `gamma` the shape constraint parameter
- `h` the kernel bandwidth smoothing parameter.
- `xx` numeric vector of x data. Missing values are not accepted.
- `zz` numeric vector of gridpoint z data. Missing values are not accepted.
- `p` degree of local polynomial used.

Value

shape constraint matrix

Author(s)

X.J. Hu

`noontemp` *Noon Temperatures in Winnipeg, Manitoba*

Description

Time Series of noon temperature observations from the Winnipeg International Airport from January 1, 1960 through December 31, 1980.

Usage

```
data(noontemp)
```

Format

A single vector.

`numericalDerivative` *Numerical Derivative of Smooth Function*

Description

Cubic spline interpolation of columns of a matrix for purpose of computing numerical derivatives at a corresponding sequence of gridpoints.

Usage

```
numericalDerivative(x, g, k, delta=.001)
```

Arguments

`x`	numeric vector
`g`	numeric-valued function of x
`k`	number of derivatives to be computed
`delta`	denominator of Newton quotient approximation

Value

numeric vector of kth derivative of g(x)

Author(s)

W.J. Braun

`relsharpen` *Ridge/Enet/LASSO Sharpening via the penalty matrix.*

Description

This is a data sharpening function to remove roughness, prior to use in local polynomial regression.

Usage

```
relsharpen(x, y, h, alpha, p=2, M=51)
```

Arguments

`x`	numeric vector of equally spaced x data. Missing values are not accepted.
`y`	vector of y data. Missing values are not accepted.
`h`	the kernel bandwidth smoothing parameter.
`alpha`	the elasticnet mixing parameter vector, with alpha in [0,1].
`p`	the order of the polynomial regression.
`M`	the length of the constraint points.

Details

Note that the predictor values are assumed to be equally spaced.

Value

numeric matrix of sharpened responses, with each column corresponding to different values of alpha

Author(s)

D.Wang

Examples

```
x<-seq(0,10,length=100)
g <- function(x) sin(x)
y<-g(x)+rnorm(100)
ys<-relsharpen(x, y, dpill(x,y), alpha=c(0.2,0.8), p=2, M=51)
y.lp2<-locpoly(x,ys[,1],bandwidth=dpill(x,y),degree=1,gridsize=100)
y.lp8<-locpoly(x,ys[,2],bandwidth=dpill(x,y),degree=1,gridsize=100)
y.lp<-locpoly(x,y,bandwidth=dpill(x,y),degree=1,gridsize=100)
curve(g,x,xlim=c(0,10))
lines(y.lp2,col=2)
lines(y.lp8,col=3)
lines(y.lp,col=5)
norm(as.matrix(g(x) - y.lp2$y),type="2")
norm(as.matrix(g(x) - y.lp8$y),type="2")
norm(as.matrix(g(x) - y.lp$y),type="2")
```

RELsharpening — *Ridge/Enet/LASSO Sharpening via the mean/local polynomial regression with large bandwidth/linear regression.*

Description

This is a function to shrink responses towards their mean/estimations of local polynomial regression with large bandwidth/estimations of linear regression as a form of data sharpening to remove roughness, and reduce the bias (when "combine=TRUE"), prior to use in local polynomial regression.

Usage

```
RELsharpening(x,y,alpha,type,bigh,hband,combine)
```

Arguments

x	numeric vector of equally spaced x data. Missing values are not accepted.
y	vector of y data. Missing values are not accepted.
alpha	the elasticnet mixing parameter vector, with alpha in [0,1].

type	The type of the base line. In total, we have three types: "mean", "big_h", and "linear".
bigh	the kernel bandwidth smoothing parameter.
hband	the kernel bandwidth smoothing parameter, which will be used in the residual sharpening method.
combine	Should the smoother combined with residual method or not, default=FALSE.

Details

Note that the predictor values are assumed to be equally spaced.

Value

numeric matrix of sharpened responses, with each column corresponding to different values of alpha

Author(s)

D.Wang

Examples

```
x<-seq(0,10,length=100)
g <- function(x) sin(x)
y<-g(x)+rnorm(100)
ys<-RELsharpening(x, y,alpha=c(0.2,0.8),"big_h", dpill(x,y)*4, dpill(x,y),combine=TRUE)
y.lp2<-locpoly(x,ys[,1],bandwidth=dpill(x,y),degree=1,gridsize=100)
y.lp8<-locpoly(x,ys[,2],bandwidth=dpill(x,y),degree=1,gridsize=100)
y.lp<-locpoly(x,y,bandwidth=dpill(x,y),degree=1,gridsize=100)
curve(g,x,xlim=c(0,10))
lines(y.lp2,col=2)
lines(y.lp8,col=3)
lines(y.lp,col=5)
norm(as.matrix(g(x) - y.lp2$y),type="2")
norm(as.matrix(g(x) - y.lp8$y),type="2")
norm(as.matrix(g(x) - y.lp$y),type="2")
```

---

relsharp_bigh *Ridge/Enet/LASSO Sharpening via the local polynomial regression with large bandwidth.*

Description

This is a function to shrink responses towards their estimations of local polynomial regression with large bandwidth as a form of data sharpening to remove roughness, prior to use in local polynomial regression.

Usage

```
relsharp_bigh(x, y, alpha, bigh)
```

Arguments

x	numeric vector of equally spaced x data. Missing values are not accepted.
y	vector of y data. Missing values are not accepted.
alpha	the elasticnet mixing parameter vector, with alpha in [0,1].
bigh	the kernel bandwidth smoothing parameter.

Details

Note that the predictor values are assumed to be equally spaced.

Value

numeric matrix of sharpened responses, with each column corresponding to different values of alpha

Author(s)

D.Wang

Examples

```
x<-seq(0,10,length=100)
g <- function(x) sin(x)
y<-g(x)+rnorm(100)
ys<-relsharp_bigh(x, y,alpha=c(0.2,0.8), dpill(x,y)*4)
y.lp2<-locpoly(x,ys[,1],bandwidth=dpill(x,y),degree=1,gridsize=100)
y.lp8<-locpoly(x,ys[,2],bandwidth=dpill(x,y),degree=1,gridsize=100)
y.lp<-locpoly(x,y,bandwidth=dpill(x,y),degree=1,gridsize=100)
curve(g,x,xlim=c(0,10))
lines(y.lp2,col=2)
lines(y.lp8,col=3)
lines(y.lp,col=5)
norm(as.matrix(g(x) - y.lp2$y),type="2")
norm(as.matrix(g(x) - y.lp8$y),type="2")
norm(as.matrix(g(x) - y.lp$y),type="2")
```

---

relsharp_bigh_c — *Ridge/Enet/LASSO Sharpening via the local polynomial regression with large bandwidth and then applying the residual sharpening method.*

Description

This is a function to shrink responses towards their estimations of local polynomial regression with large bandwidth and then apply residual sharpening as a form of data sharpening to remove roughness, prior to use in local polynomial regression.

Usage

```
relsharp_bigh_c(x, y, alpha, bigh, hband)
```

Arguments

x	numeric vector of equally spaced x data. Missing values are not accepted.
y	vector of y data. Missing values are not accepted.
alpha	the elasticnet mixing parameter vector, with alpha in [0,1].
bigh	the kernel bandwidth smoothing parameter.
hband	the kernel bandwidth smoothing parameter, which will be used in the residual sharpening method.

Details

Note that the predictor values are assumed to be equally spaced.

Value

numeric matrix of sharpened responses, with each column corresponding to different values of alpha

Author(s)

D.Wang

Examples

```
x<-seq(0,10,length=100)
g <- function(x) sin(x)
y<-g(x)+rnorm(100)
ys<-relsharp_bigh_c(x, y,alpha=c(0.2,0.8), dpill(x,y)*4, dpill(x,y))
y.lp2<-locpoly(x,ys[,1],bandwidth=dpill(x,y),degree=1,gridsize=100)
y.lp8<-locpoly(x,ys[,2],bandwidth=dpill(x,y),degree=1,gridsize=100)
y.lp<-locpoly(x,y,bandwidth=dpill(x,y),degree=1,gridsize=100)
curve(g,x,xlim=c(0,10))
lines(y.lp2,col=2)
lines(y.lp8,col=3)
lines(y.lp,col=5)
norm(as.matrix(g(x) - y.lp2$y),type="2")
norm(as.matrix(g(x) - y.lp8$y),type="2")
norm(as.matrix(g(x) - y.lp$y),type="2")
```

relsharp_linear *Ridge/Enet/LASSO Sharpening via the linear regression.*

Description

This is a function to shrink responses towards their estimations of linear regression as a form of data sharpening to remove roughness, prior to use in local polynomial regression.

Usage

```
relsharp_linear(x, y, alpha)
```

Arguments

x	numeric vector of equally spaced x data. Missing values are not accepted.
y	vector of y data. Missing values are not accepted.
alpha	the elasticnet mixing parameter vector, with alpha in [0,1].

Details

Note that the predictor values are assumed to be equally spaced.

Value

numeric matrix of sharpened responses, with each column corresponding to different values of alpha

Author(s)

D.Wang

Examples

```
x<-seq(0,10,length=100)
g <- function(x) sin(x)
y<-g(x)+rnorm(100)
ys<-relsharp_linear(x, y,alpha=c(0.2,0.8))
y.lp2<-locpoly(x,ys[,1],bandwidth=dpill(x,y),degree=1,gridsize=100)
y.lp8<-locpoly(x,ys[,2],bandwidth=dpill(x,y),degree=1,gridsize=100)
y.lp<-locpoly(x,y,bandwidth=dpill(x,y),degree=1,gridsize=100)
curve(g,x,xlim=c(0,10))
lines(y.lp2,col=2)
lines(y.lp8,col=3)
lines(y.lp,col=5)
norm(as.matrix(g(x) - y.lp2$y),type="2")
norm(as.matrix(g(x) - y.lp8$y),type="2")
norm(as.matrix(g(x) - y.lp$y),type="2")
```

relsharp_linear_c — *Ridge/Enet/LASSO Sharpening via the linear regression and then applying the residual sharpening method.*

Description

This is a function to shrink responses towards their estimations of linear regression and then apply residual sharpening as a form of data sharpening to remove roughness, prior to use in local polynomial regression.

Usage

```
relsharp_linear_c(x, y, alpha, hband)
```

Arguments

x	numeric vector of equally spaced x data. Missing values are not accepted.
y	vector of y data. Missing values are not accepted.
alpha	the elasticnet mixing parameter vector, with alpha in [0,1].
hband	the kernel bandwidth smoothing parameter, which will be used in the residual sharpening method.

Details

Note that the predictor values are assumed to be equally spaced.

Value

numeric matrix of sharpened responses, with each column corresponding to different values of alpha

Author(s)

D.Wang

Examples

```
x<-seq(0,10,length=100)
g <- function(x) sin(x)
y<-g(x)+rnorm(100)
ys<-relsharp_linear_c(x, y,alpha=c(0.2,0.8),dpill(x,y))
y.lp2<-locpoly(x,ys[,1],bandwidth=dpill(x,y),degree=1,gridsize=100)
y.lp8<-locpoly(x,ys[,2],bandwidth=dpill(x,y),degree=1,gridsize=100)
y.lp<-locpoly(x,y,bandwidth=dpill(x,y),degree=1,gridsize=100)
curve(g,x,xlim=c(0,10))
lines(y.lp2,col=2)
lines(y.lp8,col=3)
lines(y.lp,col=5)
norm(as.matrix(g(x) - y.lp2$y),type="2")
```

```
norm(as.matrix(g(x) - y.lp8$y),type="2")
norm(as.matrix(g(x) - y.lp$y),type="2")
```

---

relsharp_mean *Ridge/Enet/LASSO Sharpening via the Mean*

Description

This is a function to shrink responses towards their mean as a form of data sharpening to remove roughness, prior to use in local polynomial regression.

Usage

```
relsharp_mean(y, alpha)
```

Arguments

y	vector of y data. Missing values are not accepted.
alpha	The elasticnet mixing parameter vector, with alpha in [0,1].

Details

Note that the predictor values are assumed to be equally spaced.

Value

numeric matrix of sharpened responses, with each column corresponding to different values of alpha

Author(s)

D.Wang

Examples

```
x<-seq(0,10,length=100)
g <- function(x) sin(x)
y<-g(x)+rnorm(100)
ys<-relsharp_mean(y,alpha=c(0.2,0.8))
y.lp2<-locpoly(x,ys[,1],bandwidth=dpill(x,y),degree=1,gridsize=100)
y.lp8<-locpoly(x,ys[,2],bandwidth=dpill(x,y),degree=1,gridsize=100)
y.lp<-locpoly(x,y,bandwidth=dpill(x,y),degree=1,gridsize=100)
curve(g,x,xlim=c(0,10))
lines(y.lp2,col=2)
lines(y.lp8,col=3)
lines(y.lp,col=5)
norm(as.matrix(g(x) - y.lp2$y),type="2")
norm(as.matrix(g(x) - y.lp8$y),type="2")
norm(as.matrix(g(x) - y.lp$y),type="2")
```

relsharp_mean_c *Ridge/Enet/LASSO Sharpening via the Mean and then applying the residual sharpening method.*

Description

This is a function to shrink responses towards their mean and then apply residual sharpening as a form of data sharpening to remove roughness, prior to use in local polynomial regression.

Usage

```
relsharp_mean_c(x, y, alpha, hband)
```

Arguments

x	numeric vector of equally spaced x data. Missing values are not accepted.
y	vector of y data. Missing values are not accepted.
alpha	the elasticnet mixing parameter vector, with alpha in [0,1].
hband	the kernel bandwidth smoothing parameter, which will be used in the residual sharpening method.

Details

Note that the predictor values are assumed to be equally spaced.

Value

numeric matrix of sharpened responses, with each column corresponding to different values of alpha

Author(s)

D.Wang

Examples

```
x<-seq(0,10,length=100)
g <- function(x) sin(x)
y<-g(x)+rnorm(100)
ys<-relsharp_mean_c(x, y,alpha=c(0.2,0.8), dpill(x,y))
y.lp2<-locpoly(x,ys[,1],bandwidth=dpill(x,y),degree=1,gridsize=100)
y.lp8<-locpoly(x,ys[,2],bandwidth=dpill(x,y),degree=1,gridsize=100)
y.lp<-locpoly(x,y,bandwidth=dpill(x,y),degree=1,gridsize=100)
curve(g,x,xlim=c(0,10))
lines(y.lp2,col=2)
lines(y.lp8,col=3)
lines(y.lp,col=5)
norm(as.matrix(g(x) - y.lp2$y),type="2")
norm(as.matrix(g(x) - y.lp8$y),type="2")
norm(as.matrix(g(x) - y.lp$y),type="2")
```

testfun *Functions for Testing Purposes*

Description

Functions that can be used in simulations to test the effectiveness of the sharpening procedures.

Usage

```
testfun(x, k)
```

Arguments

x	numeric vector
k	a numeric constant that controls the height of the peak of the test function; if missing, a periodic function is supplied

Value

numeric vector of function output

Author(s)

D.Wang

Index