Package ‘sharpData’

August 19, 2019

Title Data Sharpening

Version 1.2

Author W.J. Braun

Description Functions and data sets inspired by data sharpening -
data perturbation to achieve improved performance in
nonparametric estimation, as described in Choi, E., Hall, P.
and Rousson, V. (2000) <doi:10.1214/aos/1015957396>.
Capabilities for enhanced local linear regression function
and derivative estimation are included, as well as an
asymptotically correct iterated data sharpening estimator
for any degree of local polynomial regression estimation.
A cross-validation-based bandwidth selector is included which,
in concert with the iterated sharpener, will often provide
superior performance, according to a median integrated squared
error criterion. Sample data sets are provided to illustrate
function usage.

Maintainer W.J. Braun <john.braun@ubc.ca>

LazyLoad true

LazyData true

Depends R (>= 2.0.1), KernSmooth, stats, quadprog

ZipData no

License Unlimited

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-08-19 18:50:09 UTC

R topics documented:

`burnRate` *Firebrand Burning Properties*

Description

The `burnRate` data frame contains laboratory data on the proportion of remaining fuel in a piece of wood that has burned for a fixed period of time subjected to a fixed windspeed.

Usage

```
data(burnRate)
```

Format

This data frame contains the following columns:

proportionBurned a numeric vector

densityRatio ratio of windspeed, multiplied by density of air, to density of firebrand

Source

Albini, F. USDA Forest Service General Technical Report INT-56, 1979.

`CVsharp` *Cross-Validation Bandwidth Selector for Local Polynomial Regression*

Description

Cross-validation bandwidth selector for iterated sharpened responses for bias reduction in function estimation.

Usage

```
CVsharp(x, y, deg, nsteps)
```

Arguments

`x`	a numeric vector containing the predictor variable values.
`y`	a numeric vector containing the response variable values.
`deg`	a numeric vector containing the local polynomial degree used.
`nsteps`	a numeric vector containing the number of iteration steps.

Details

If nsteps is specified to be 0, then the CV bandwidth for conventional local polynomial regression is provided.

Value

a list containing 3 elements: the candidate bandwidths; the corresponding CV scores; the selected optimal bandwidth.

Author(s)

W.J. Braun

See Also

`locpoly`

Examples

```
speed <- MPG[, 1]
mpg <- MPG[, 2]
h <- CVsharp(speed, mpg, 0, 0)$CVh # conventional local constant regression bandwidth
mpg.l0 <- locpoly(speed, mpg, bandwidth=h, degree=0)
h <- CVsharp(speed, mpg, 0, 1)$CVh # 1-sharpened local constant regression bandwidth
mpgSharp <- sharpiteration(speed, mpg, 0, h, 1)
mpg.l1 <- locpoly(speed, mpgSharp[[1]], bandwidth=h, degree=0)
h <- CVsharp(speed, mpg, 0, 5)$CVh # 5-sharpened local constant regression bandwidth
mpgSharp <- sharpiteration(speed, mpg, 0, h, 5)
mpg.l5 <- locpoly(speed, mpgSharp[[5]], bandwidth=h, degree=0)
plot(mpg ~ speed)
lines(mpg.l0)  # unsharpened function estimation
lines(mpg.l1, col=2, lty=2)  # sharpened function estimation (1 steps)
lines(mpg.l5, col=4, lty=3)  # sharpened function estimation (5 steps)
```

LL.fit *Local linear Regression Estimation*

Description

This computes the local linear regression estimator at a few gridpoints. For a large number of gridpoints, it is not efficient.

Usage

```
LL.fit(x, y, xgrid, h, kernel)
```

Arguments

`x`	numeric explanatory vector
`y`	numeric response vector
`xgrid`	numeric vector
`h`	numeric bandwidth
`kernel`	character constant

Value

a vector of predictor values

Author(s)

W.J. Braun

References

Wand and Jones, 1996, Kernel Smoothing, Chapman and Hall.

See Also

`locpoly`

Examples

```
x <- sort(runif(100))
y <- x + sin(x) + rnorm(100)*.1
LL.fit(x,y,c(.3, .5, .7),h=.5)
```

LLmat *Local Linear Regression Matrix*

Description

This computes the smoother matrix for the monotonic local linear sharpening problem.

Usage

```
LLmat(xgrid, x,  h, kernel)
```

Arguments

`xgrid`	numeric vector
`x`	numeric explanatory vector
`h`	numeric bandwidth
`kernel`	character constant

Value

a list containing the A matrix and the number of rows in A.

Author(s)

W.J. Braun

LLmono *Monotonized Local Linear Regression*

Description

Locally linear regression is applied to bivariate data. The response is 'sharpened' or perturbed in a way to render a monotonically increasing curve estimate.

Usage

```
LLmono(x, y, h, xgrid, numgrid = 401, kernel="biweight", call.plot =
TRUE, ...)
```

Arguments

`x`	a vector of explanatory variable observations
`y`	binary vector of responses
`h`	bandwidth
`xgrid`	gridpoints on x-axis where estimates are taken
`numgrid`	number of equally-spaced gridpoints (if xgrid not specified)
`kernel`	character constant
`call.plot`	if TRUE (default), the data, sharpened data and estimated curve are plotted.
`...`	other arguments for plot

Details

Data are perturbed the smallest possible L2 distance subject to the constraint that the local linear estimate is monotonically increasing.

Value

`x`	original explanatory variable
`y`	original responses
`ysharp`	sharpened responses
`h`	bandwidth
`xgrid`	gridpoints
`ygrid`	sharpened curve estimate

Author(s)

W.J.Braun

References

Braun, W.J. and Hall, P., Data Sharpening for Nonparametric Estimation Subject to Constraints, Journal of Computational and Graphical Statistics, 2001

Examples

```
gridpts <- seq(1, 10, length=101)
x <- seq(1, 10, length=51)
p <- exp(-1 + .2*x)/(1 + exp(-1 + .2*x))
y <- rbinom(51, 1, p)
LLmono(x, y, h=0.6, xgrid=gridpts)
lines(x,p) # true mean response
```

LLsharpen — *Data Sharpening for Local Linear Regression*

Description

Calculation of sharpened responses for bias reduction in function and first derivative estimation, assuming a gaussian kernel is used in bivariate scatterplot smoothing.

Usage

```
LLsharpen(x, y, h)
```

Arguments

x	a numeric vector containing the predictor variable values.
y	a numeric vector containing the response variable values.
h	a numeric vector containing the (scalar) bandwidth.

Value

a vector containing the sharpened (i.e. perturbed) response values, ready for input into a local linear regression estimator.

Author(s)

W.J. Braun

See Also

`locpoly`

Examples

```
speed <- MPG[, 1]
mpg <- MPG[, 2]
h <- dpill(speed, mpg)*2
mpgSharp <- LLsharpen(speed, mpg, h)
mpg.lS <- locpoly(speed, mpgSharp, bandwidth=h, drv=1, degree=1)
mpg.lX <- locpoly(speed, mpg, bandwidth=h, drv=1, degree=1)
plot(mpg.lX, type="l")  # unsharpened derivative estimation
lines(mpg.lS, col=2, lty=2)  # sharpened derivative estimation
```

`MPG` *Mileage Data*

Description

The `MPG` data frame has 15 rows and 10 columns.

Usage

```
data(MPG)
```

Format

This data frame contains the following columns:

speed a numeric vector of cruising speeds in miles per hour

corsica88 miles per gallon for a 1988 Corsica

legacy93 miles per gallon for a 1993 Legacy

olds94 miles per gallon for a 1994 Oldsmobile

cutlass94 miles per gallon for a 1994 Oldsmobile Cutlass

chevpickup94 miles per gallon for a 1994 Chevrolet Pickup

cherokee94 miles per gallon for a 1994 Jeep Cherokee

villager94 miles per gallon for a 1994 Villager

prizm95 miles per gallon for a 1995 Prizm

celica97 miles per gallon for a 1997 Toyota Celica

Source

B.H. West, R.N. McGill, J.W. Hodgson, S.S. Sluder, D.E. Smith, Development and Verification of Light-Duty Modal Emissions and Fuel Consumption Values for Traffic Models, Washington, DC, April 1997, and additional project data, April 1998.

Examples

```
data(MPG)
plot(celica97 ~ speed, data = MPG)
```

`NW.fit` *Nadaraya-Watson Regression Estimation*

Description

This computes the Nadaraya-Watson local constant regression estimator at a single point. For multiple points, it is not efficient.

Usage

```
NW.fit(x, y, xgrid, h, kernel)
```

Arguments

`x`	numeric explanatory vector
`y`	numeric response vector
`xgrid`	numeric vector
`h`	numeric bandwidth
`kernel`	character constant

Value

a vector of predictor values

Author(s)

W.J. Braun

References

Wand and Jones, 1996, Kernel Smoothing, Chapman and Hall.

See Also

`locpoly`

Examples

```
x <- sort(runif(100))
y <- x + sin(x) + rnorm(100)*.1
NW.fit(x,y,c(.3, .5, .7),h=.1)
```

NWmat *Smoother Matrix for Nadaraya-Watson Local Constant Regression*

Description

This function computes the A matrix for use in quadprog. This matrix corresponds to the derivative matrix for constrained sharpening in the monotonic regression problem.

Usage

```
NWmat(xgrid, x, h, kernel = "biweight")
```

Arguments

`xgrid`	a vector of gridpoints to evaluate estimate at
`x`	a vector containing the predictor data
`h`	a scalar bandwidth
`kernel`	a character string naming the kernel function

Value

a matrix

Author(s)

W.J. Braun

NWmono *Monotonized Nadaraya-Watson Regression*

Description

Nadaraya-Watson or locally constant regression is applied to bivariate data. The response is 'sharpened' or perturbed in a way to render a monotonically increasing curve estimate.

Usage

```
NWmono(x, y, h, xgrid, numgrid = 401, kernel="biweight", call.plot =
TRUE, ...)
```

Arguments

`x`	a vector of explanatory variable observations
`y`	binary vector of responses
`h`	bandwidth
`xgrid`	gridpoints on x-axis where estimates are taken
`numgrid`	number of equally-spaced gridpoints (if xgrid not specified)
`kernel`	character constant
`call.plot`	if TRUE (default), the data, sharpened data and estimated curve are plotted.
`...`	other arguments for plot

Details

Data are perturbed the smallest possible L2 distance subject to the constraint that the Nadaraya-Watson estimate is monotonically increasing.

Value

`x`	original explanatory variable
`y`	original responses
`ysharp`	sharpened responses
`h`	bandwidth
`xgrid`	gridpoints
`ygrid`	sharpened curve estimate

Author(s)

W.J.Braun

References

Braun, W.J. and Hall, P., Data Sharpening for Nonparametric Estimation Subject to Constraints, Journal of Computational and Graphical Statistics, 2001

Examples

```
gridpts <- seq(1, 10, length=101)
x <- seq(1, 10, length=51)
p <- exp(-1 + .2*x)/(1 + exp(-1 + .2*x))
y <- rbinom(51, 1, p)
NWmono(x, y, h=0.6, xgrid=gridpts)
lines(x,p) # true mean response
```

sharpiteration — *Iterated Data Sharpening for Local Polynomial Regression*

Description

Calculation of sharpened responses for bias reduction in function and estimation, assuming a gaussian kernel is used in bivariate scatterplot smoothing.

Usage

```
sharpiteration(x, y, deg, h, nsteps, na.rm, ...)
```

Arguments

`x`	a numeric vector containing the predictor variable values.
`y`	a numeric vector containing the response variable values.
`deg`	a numeric vector containing the local polynomial degree used.
`h`	a numeric vector containing the (scalar) bandwidth.
`nsteps`	a numeric vector containing the number of iteration steps.
`na.rm`	a logical value indicating whether to remove missing values from fitted vectors
`...`	additional arguments to locpoly

Value

a list with elements containing the sharpened (i.e. perturbed) response values, ready for input into a local polynomial regression estimator. The ith list element corresponds to i steps of data sharpening.

Author(s)

W.J. Braun

See Also

`locpoly`

Examples

```
speed <- MPG[, 1]
mpg <- MPG[, 2]
h <- dpill(speed, mpg)
mpgSharp <- sharpiteration(speed, mpg, 1, h, 2)
mpg.lS <- locpoly(speed, mpgSharp[[2]], bandwidth=h, degree=1)
mpg.lX <- locpoly(speed, mpg, bandwidth=h, degree=1)
plot(mpg ~ speed)
lines(mpg.lX)  # unsharpened function estimation
lines(mpg.lS, col=2, lty=2)  # sharpened function estimation (2 steps)
```

whale *Whale data*

Description

Nursing times for a baby beluga whale.

Usage

```
data(whale)
```

Format

A data frame with 228 observations on the following 3 variables.

V1 a numeric vector

V2 a numeric vector

V3 a factor with levels `0` `104` `118` `119` `126` `127` `132` `135` `137` `14` `144` `146` `150` `151` `153` `156` `157` `160` `166` `167` `168` `169` `170` `171` `172` `174` `175` `176` `180` `186` `187` `189` `191` `192` `193` `196` `197` `198` `199` `200` `204` `205` `216` `218` `222` `223` `225` `226` `228` `229` `230` `231` `232` `236` `239` `243` `244` `247` `252` `253` `255` `257` `260` `267` `271` `274` `275` `277` `284` `285` `286` `288` `291` `292` `299` `308` `320` `323` `326` `332` `338` `339` `340` `344` `345` `349` `351` `353` `354` `359` `360` `362` `371` `372` `377` `380` `386` `404` `409` `411` `419` `423` `426` `429` `430` `432` `433` `435` `438` `440` `441` `442` `443` `444` `445` `446` `449` `450` `453` `456` `462` `463` `464` `470` `473` `477` `48` `485` `491` `492` `494` `495` `497` `504` `506` `509` `51` `513` `515` `524` `528` `533` `537` `538` `541` `565` `579` `59` `590` `600` `605` `613` `644` `648` `659` `68` `688` `69` `693` `694` `702` `714` `72` `720` `737` `74` `750` `756` `772` `80` `805` `813` `825` `84` `85` `870` `873` `888` `92` `93` `954` `96` `98` `M`

Source

Simonoff, J. Smoothing Methods in Statistics, Springer, 1996.

Index