

Package: sharpData (via r-universe)

October 12, 2026

Version 1.4

Title Data Sharpening

Author W. John Braun <john.braun@ubc.ca>

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

Depends R (>= 3.5.0), KernSmooth, stats, quadprog

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). 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.

LazyLoad true

LazyData true

ZipData no

License Unlimited

NeedsCompilation yes

Repository <https://wjbraun.r-universe.dev>

Date/Publication 2021-03-30 06:40:02 UTC

RemoteUrl <https://github.com/cran/sharpData>

RemoteRef HEAD

RemoteSha e0e336ee024041f7b8d7b29e477fa279dbdc95c8

Contents

burnRate 2

2

burnRate

CVsharp

LLsharpen

Monolpoly

MonoMat

MPG

sharpiteration

whale

Index

3

4

5

6

7

8

9

10

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
- species** factor listing tree species
- diameter** numeric vector of diameter of burned particle in cm
- windspeed** windspeed in cm per second
- testTime** length of test in seconds

Source

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

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.l15 <- 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)

```

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

References

Choi, E., Hall, P. and Rousson, V. (2000) Data sharpening methods for bias reduction in nonparametric regression. *Annals of Statistics* 28(5) 1339-1355.

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

```

Monolpoly

Monotonized Local Regression

Description

Local constant and local linear regression are applied to bivariate data. The response is ‘sharpened’ or perturbed in a way to render a monotonically increasing curve estimate.

Usage

```
Monolpoly(x, y, h, d=1, xgrid, numgrid = 401, ...)
```

Arguments

x	a vector of explanatory variable observations
y	binary vector of responses
h	bandwidth
d	degree, can be either 0 or 1
xgrid	gridpoints on x-axis where monotonicity constraint is enforced
numgrid	number of equally-spaced gridpoints (if xgrid not specified)
...	other arguments for locpoly

Details

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

Value

x	locations of function estimate evaluations
y	function estimate evaluations (sharpened - monotonized)
ysharp	sharpened responses

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)
plot(x, y)
lines(Monolpoly(x, y, h=0.6, xgrid=gridpts))
##
plot(faithful)
with(faithful,
lines(Monolpoly(eruptions, waiting, h=0.1, d=1,
range=c(1.55, 5.15))))
```

MonoMat

Matrix of derivative coefficients for local polynomial estimates

Description

This computes a matrix of coefficients of the first derivatives of monotonic local linear sharpening problem.

Usage

```
MonoMat(xgrid, x, h, d)
```

Arguments

xgrid	numeric vector of locations where monotonicity constraint is to be enforced
x	numeric explanatory vector
h	numeric bandwidth
d	local polynomial degree, can be either 0 or 1

Value

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

Author(s)

W.J. Braun

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)
```

sharpiteration	<i>Iterated Data Sharpening for Local Polynomial Regression</i>
----------------	---

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 *i*th 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	<i>Whale data</i>
-------	-------------------

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

* **datasets**

burnRate, [2](#)

MPG, [7](#)

whale, [9](#)

* **models**

CVsharp, [3](#)

LLsharpen, [4](#)

Monopoly, [5](#)

MonoMat, [6](#)

sharpiteration, [8](#)

burnRate, [2](#)

CVsharp, [3](#)

LLsharpen, [4](#)

Monopoly, [5](#)

MonoMat, [6](#)

MPG, [7](#)

sharpiteration, [8](#)

whale, [9](#)