Package 'OSCV'

January 20, 2025

Title One-Sided Cross-Validation

Version 1.0

Author Olga Savchuk

Maintainer Olga Savchuk <olga.y.savchuk@gmail.com>

Description Functions for implementing different versions of the OSCV method in the kernel regression and density estimation frameworks. The package mainly supports the following articles: (1) Savchuk, O.Y., Hart, J.D. (2017). Fully robust one-sided cross-validation for regression functions. Computational Statistics, <doi:10.1007/s00180-017-0713-7> and (2) Savchuk, O.Y. (2017). One-sided cross-validation for nonsmooth density functions, <arXiv:1703.05157>.

Depends R (>= 3.1.1), mc2d

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-03-18 17:51:59 UTC

Contents

`ASE_reg` *The ASE function for the local linear estimator (LLE) in the regression context.*

Description

Computing $ASE(h)$, the value of the ASE function for the local linear estimator in the regression context, for the given vector of h values.

Usage

```
ASE_reg(h, desx, y, rx)
```

Arguments

`h`	numerical vector of bandwidth values,
`desx`	numerical vecror of design points,
`y`	numerical vecror of data points corresponding to the design points $desx$,
`rx`	numerical vecror of values of the regression function at $desx$.

Details

The average squared error (ASE) is used as a measure of performace of the local linear estimator based on the Gaussian kernel.

Value

The vector of values of $ASE(h)$ for the correponsing vector of h values.

References

Hart, J.D. and Yi, S. (1998) One-sided cross-validation. *Journal of the American Statistical Association*, 93(442), 620-631.

See Also

`loclin`, `h_ASE_reg`, `CV_reg`, `OSCV_reg`.

Examples

```
## Not run:
# Example (ASE function for a random sample of size n=100 generated from the function reg3 that
# has six cusps. The function originates from the article of Savchuk et al. (2013).
# The level of the added Gaussian noise is sigma=1/1000).
n=100
dx=(1:n-0.5)/n
regx=reg3(dx)
ydat=regx+rnorm(n,sd=1/1000)
harray=seq(0.003,0.05,len=300)
ASEarray=ASE_reg(harray,dx,ydat,regx)
hmin=round(h_ASE_reg(dx,ydat,regx),digits=4)
dev.new()
plot(harray,ASEarray,'l',lwd=3,xlab="h",ylab="ASE",main="ASE function for a random sample
from r3",cex.lab=1.7,cex.axis=1.7,cex.main=1.5)
legend(0.029,0.0000008,legend=c("n=100","sigma=1/1000"),cex=1.7,bty="n")
legend(0.005,0.000002,legend=paste("h_ASE=",hmin),cex=2,bty="n")

## End(Not run)
```

CV_reg *The cross-validation (CV) function in the regression context.*

Description

Computing $CV(h)$, the value of the CV function in the regression context.

Usage

```
CV_reg(h, desx, y)
```

Arguments

h	numerical vector of bandwidth values,
desx	numerical vecror of design points,
y	numerical vecror of data values corresponding to the design points $desx$.

Details

The CV function is a measure of fit of the regression estimate to the data. The local linear estimator based on the Gaussian kernel is used. The cross-validation bandwidth is the minimizer of the CV function.

Value

The vector of values of $CV(h)$ for the correponsing vector of h values.

References

Stone, C.J. (1977) Consistent nonparametric regression. *Annals of Statistics*, 5(4), 595-645.

See Also

loclin, h_ASE_reg, ASE_reg, OSCV_reg.

Examples

```
## Not run:
# Example (Old Faithful geyser). Take x=waiting time; y=eruption duration. The sample size n=272.
xdat=faithful[[2]]
ydat=faithful[[1]]
harray=seq(0.5,10,len=100)
cv=CV_reg(harray,xdat,ydat)
R=range(xdat)
h_cv=round(optimize(CV_reg,c(0.01,(R[2]-R[1]/4)),desx=xdat,y=ydat)$minimum,digits=4)
dev.new()
plot(harray,cv,'l',lwd=3,xlab="h",ylab="CV(h)",main="CV function for the Old Faithful
geyser data", cex.lab=1.7,cex.axis=1.7,cex.main=1.5)
legend(6,0.155,legend="n=272",cex=1.8,bty="n")
legend(1,0.18,legend=paste("h_CV=",h_cv),cex=2,bty="n")

## End(Not run)
```

C_smooth *The OSCV smooth rescaling constant.*

Description

Computing the OSCV smooth rescaling constant that corresponds to using the two-sided kernel H_I for the cross-validation purposes and the Gaussian kernel in the estimation stage. The constant is applicable for the OSCV versions in the regression and kernel density estimation contexts.

Usage

```
C_smooth(alpha, sigma)
```

Arguments

alpha	first parameter of the two-sided cross-validation kernel H_I,
sigma	second parameter of the two-sided cross-validation kernel H_I.

Details

Computation of the OSCV rescaling constant C (see (10) in Savchuk and Hart (2017) or (3) in Savchuk (2017)). The constant is a function of the parameters (α, σ) of the two-sided cross-validation kernel H_I defined by expression (15) in Savchuk and Hart (2017). The Gaussian kernel is used for computing the ultimate (regression or density) estimate. The constant is used in the OSCV versions for kernel regression and density estimation. Notice that in the cases $\alpha = 0$, $\sigma > 0$ and $\sigma = 1$, $-\infty < \alpha < \infty$ the kernel H_I reduces to the Gaussian kernel.

Value

The OSCV smooth rescaling constant C for the given values of the parameters α and σ.

References

- Savchuk, O.Y., Hart, J.D. (2017). Fully robust one-sided cross-validation for regression functions. *Computational Statistics*, doi:10.1007/s00180-017-0713-7.
- Savchuk, O.Y. (2017). One-sided cross-validation for nonsmooth density functions, arXiv:1703.05157.

See Also

L_I, H_I, OSCV_reg, h_OSCV_reg, OSCV_LI_dens, OSCV_Gauss_dens, h_OSCV_dens, loclin.

Examples

```
# OSCV rescaling constant for the robust cross-validation kernel with
# (alpha,sigma)=(16.8954588,1.01).
C_smooth(16.8954588,1.01)
# OSCV smooth rescaling constant in the case when the kernel H_I is Gaussian.
C_smooth(1,1)
```

---

fstar *Nonsmooth density function with seven cusps.*

Description

Nonsmooth density f^* with seven cusps introduced in the article of Savchuk (2017).

Usage

```
fstar(u)
```

Arguments

u	numerical vecror of argument values in the range [-3,3].

Details

The function f^* consists of straight lines with different slopes connected together. The support of the density is [-3,3].

Value

The vector of values of f^* corresponding to the values of the vector u.

References

Savchuk, O.Y. (2017). One-sided cross-validation for nonsmooth density functions, arXiv:1703.05157.

See Also

sample_fstar, ISE_fstar.

Examples

```
## Not run:
dev.new()
plot(seq(-3.5,3.5,len=1000),fstar(seq(-3.5,3.5,len=1000)),'l',lwd=3,
main="Nonsmooth density fstar with seven cusps", xlab="argument", ylab="density",cex.main=1.5,
cex.axis=1.7,cex.lab=1.7)

## End(Not run)
```

---

h_ASE_reg *The ASE-optimal bandwidth in the regression context.*

Description

Computing the ASE-optimal bandwidth for the Gaussian local linear regression estimator.

Usage

```
h_ASE_reg(desx, y, rx)
```

Arguments

desx	numerical vecror of design points,
y	numerical vecror of data points corresponding to the design points $desx$,
rx	numerical vecror of the regression function values at $desx$.

Details

Computing the ASE-optimal bandwidth for the local linear estimator in the regression context. The ASE-optimal bandwidth is the global minimizer of the ASE function ASE_reg. This bandwidth is optimal for the data set at hand.

Value

The ASE-optimal bandwidth (scalar).

See Also

ASE_reg, loclin.

Examples

```
## Not run:
# Simulated example.
n=300
dx=runif(n)            #uniform design
regx=5*dx^10*(1-dx)^2+2.5*dx^2*(1-dx)^10
ydat=regx+rnorm(n,sd=1/250)
hase=round(h_ASE_reg(dx,ydat,regx),digits=4)
u=seq(0,1,len=1000)
fun=5*u^10*(1-u)^2+2.5*u^2*(1-u)^10
dev.new()
plot(dx,ydat,pch=20,cex=1.5,xlab="argument",ylab="function",cex.lab=1.7,cex.axis=1.7,
main="Function, data, and the ASE-optimal bandwidth",cex.main=1.5)
lines(u,fun,'l',lwd=3,col="blue")
legend(0,0.03,legend=paste("h_ASE=",hase),cex=1.8,bty="n")
legend(0.6,-0.002,legend=paste("n=",n),cex=2,bty="n")

## End(Not run)
```

H_I *The family of two-sided cross-validation kernels H_I.*

Description

The family of two-sided cross-validation kernels H_I defined by equation (15) of Savchuk and Hart (2017).

Usage

```
H_I(u, alpha, sigma)
```

Arguments

u	numerical vector of argument values,
alpha	first parameter of the cross-validation kernel H_I,
sigma	second parameter of the cross-validation kernel H_I.

Details

The family of the two-sided cross-validation kernels $H_I(u;\alpha,\sigma) = (1+\alpha)\phi(u) - \alpha\phi(u/\sigma)/\sigma$, where ϕ denotes the Gaussian kernel, $-\infty < \alpha < \infty$ and $\sigma > 0$ are the parameters of the kernel. See expression (15) of Savchuk and Hart (2017). The robust kernel plotted in Figure 1 of Savchuk and Hart (2017) is obtained by setting $\alpha = 16.8954588$ and $\sigma = 1.01$. Note that the kernels H_I are also used for the bandwidth selection purposes in the indirect cross-validation (ICV) method (see

expression (4) of Savchuk, Hart, and Sheather (2010)). The kernel H_I is a two-sided analog of the one-sided kernel L_I. The Gaussian kernel ϕ is the special case of H_I obtained by either setting $\alpha = 0$ or $\sigma = 1$.

Value

The value of $H_I(u; \alpha, \sigma)$.

References

- Savchuk, O.Y., Hart, J.D. (2017). Fully robust one-sided cross-validation for regression functions. *Computational Statistics*, doi:10.1007/s00180-017-0713-7.
- Savchuk, O.Y., Hart, J.D., Sheather, S.J. (2010). Indirect cross-validation for density estimation. *Journal of the American Statistical Association*, 105(489), 415-423.

See Also

L_I, C_smooth, OSCV_reg, loclin.

Examples

```
## Not run:
# Plotting the robust kernel from Savchuk and Hart (2017) with alpha=16.8954588 and sigma=1.01.
u=seq(-5,5,len=1000)
ker=H_I(u,16.8954588,1.01)
dev.new()
plot(u,ker,'l',lwd=3,cex.axis=1.7, cex.lab=1.7)
title(main="Robust kernel H_I along with the Gaussian kernel (phi)",cex=1.7)
lines(u,dnorm(u),lty="dashed",lwd=3)
legend(-4.85,0.3,lty=c("solid","dashed"),lwd=c(3,3),legend=c("H_I","phi"),cex=1.5)
legend(1,0.4,legend=c("alpha=16.8955","sigma=1.01"),cex=1.5,bty="n")

## End(Not run)
```

---

h_OSCV_dens *The OSCV bandwidth in the density estimation context.*

Description

Computing the OSCV bandwidth for the Gaussian density estimator. The one-sided Gaussian kernel L_G is used in the bandwidth selection stage. The (anticipated) smoothness of the density function is to be specified by the user.

Usage

```
h_OSCV_dens(dat, stype)
```

Arguments

dat	numerical vecror of data values,
stype	specifies (anticipated) smoothness of the density function. Thus, $stype = 0$ corresponds to the *smooth* density, whereas $stype = 1$ corresponds to the *nonsmooth* density.

Details

Computing the OSCV bandwidth for the data vector dat. The one-sided Gaussian kernel L_G is used for the cross-validation purposes and the Gaussian kernel is used for computing the ultimate density estimate. The (anticipated) smoothness of the underlying density function is to be specified. Thus,

- $stype = 0$ corresponds to the smooth density;
- $stype = 1$ corresponds to the nonsmooth density.

It is usually assumed that the density is smooth if no preliminary information about its nonsmoothness is available. No additional rescaling of the computed bandwidth is needed. The smoothness of the density function $stype$, essentially, determines the value of the bandwidth rescaling constant that is used in the body of the function. Thus, the constant is equal to 0.6168471 in the smooth case, whereas it is equal to 0.5730 in the nonsmooth case. See Savchuk (2017) for details. The OSCV bandwidth is the minimizer of the OSCV function OSCV_Gauss_dens.

Value

The OSCV bandwidth (scalar).

References

Savchuk, O.Y. (2017). One-sided cross-validation for nonsmooth densty functions, arXiv:1703.05157.

See Also

OSCV_Gauss_dens, C_smooth, h_OSCV_reg.

Examples

```
## Not run:
data=faithful[,1]         # Data on n=272 eruption duration of the Old Faithful geyser.
harray=seq(0.025,0.6,len=100)
OSCV_array=OSCV_Gauss_dens(harray,data,0)
dev.new()
plot(harray,OSCV_array,lwd=3,'l',xlab="h",ylab="L_G-based OSCV",
main="OSCV_G(h) for the data on eruption duration",cex.main=1.5,cex.lab=1.7,cex.axis=1.7)
h_oscv=round(h_OSCV_dens(data,0),digits=4) #smoothness of the underlying density is assumed
legend(0.04,-0.25,legend=c("n=272",paste("h_OSCV=",h_oscv)),cex=2,bty="n")

## End(Not run)
```

h_OSCV_reg *The OSCV bandwidth in the regression context.*

Description

Computing the OSCV bandwidth for the Gaussian local linear regression estimator. The Gaussian kernel is used in the bandwidth selection stage. The smoothness of the regression function is to be specified by the user.

Usage

```
h_OSCV_reg(desx, y, stype)
```

Arguments

desx	numerical vecror of design points,
y	numerical vecror of data points corresponding to the design points $desx$,
stype	smoothness of the regression function: ($stype = 0$) smooth function; ($stype = 1$) nonsmooth function.

Details

Computing the OSCV bandwidth for the data vector $(desx, y)$. The Gaussian kernel is used for the cross-validation purposes and in the stage of computing the resulting local linear regression estimate. No additional rescaling of the computed bandwidth is needed. The smoothness of the regression function $stype$, essentially, determines the value of the bandwidth rescaling constant that is chosen in the body of the function. Thus, the constant is equal to 0.6168471 in the smooth case, and 0.5730 in the nonsmooth case. See Savchuk, Hart and Sheather (2016). The OSCV bandwidth is the minimizer of the OSCV function OSCV_reg.

Value

The OSCV bandwidth (scalar).

References

- Hart, J.D. and Yi, S. (1998). One-sided cross-validation. *Journal of the American Statistical Association*, 93(442), 620-631.
- Savchuk, O.Y., Hart, J.D., Sheather, S.J. (2013). One-sided cross-validation for nonsmooth regression functions. *Journal of Nonparametric Statistics*, 25(4), 889-904.
- Savchuk, O.Y., Hart, J.D., Sheather, S.J. (2016). Corrigendum to "One-sided cross-validation for nonsmooth regression functions". *Journal of Nonparametric Statistics*, 28(4), 875-877.
- Savchuk, O.Y., Hart, J.D. (2017). Fully robust one-sided cross-validation for regression functions. *Computational Statistics*, doi:10.1007/s00180-017-0713-7.

See Also

OSCV_reg, loclin, C_smooth, h_OSCV_dens, h_ASE_reg.

Examples

```
## Not run:
# Example (Old Faithful geyser)
xdat=faithful[[2]]     # waiting time
ydat=faithful[[1]]     # eruption duration
u=seq(40,100,len=1000)
h_oscv=round(h_OSCV_reg(xdat,ydat,0),digits=4)
l=loclin(u,xdat,ydat,h_oscv)
dev.new()
plot(xdat,ydat,pch=20,cex=1.5,cex.axis=1.7,cex.lab=1.7,xlab="waiting time",
ylab="eruption duration")
lines(u,l,'l',lwd=3)
title(main="Data and LLE",cex.main=1.7)
legend(35,5,legend=paste("h_OSCV=",h_oscv),cex=2,bty="n")
legend(80,3,legend="n=272",cex=2,bty="n")

## End(Not run)
```

ISE_fstar — *The ISE function in the kernel density estimation (KDE) context in the case when the underlying density is* fstar.

Description

Computing the ISE function for the Gaussian density estimator obtained from a random sample of size n generated from fstar.

Usage

```
ISE_fstar(h, n)
```

Arguments

h	numerical vector of bandwidth values,
n	sample size (number of data points generated from fstar).

Details

The integrated squared error (ISE) is a measure of closeness of the Gaussian density estimate computed from a data set generated from fstar to the true density.

Value

The vector of values of the ISE function for the correponsing vector of h values.

References

Savchuk, O.Y. (2017). One-sided cross-validation for nonsmooth density functions, arXiv:1703.05157.

See Also

fstar, sample_fstar.

Examples

```
## Not run:
dev.new()
harray=seq(0.05,1.5,len=1000)
ISEarray=ISE_fstar(harray,100)
h_ISE=round(harray[which.min(ISEarray)],digits=4)
dev.new()
plot(harray,ISEarray,lwd=3,'l',xlab="h",ylab="ISE",main="ISE(h)",cex.main=2,cex.lab=1.7,
cex.axis=1.7)
legend(0.35,ISEarray[5],legend=c("n=100",paste("h_ISE=",h_ISE)),cex=1.8,bty="n")

## End(Not run)
```

---

loclin *Computing the local linear estimate (LLE).*

Description

Computing the LLE based on data $(desx, y)$ over the given vector of the argument values u. The Gausssian kernel is used. See expression (3) in Savchuk and Hart (2017).

Usage

```
loclin(u, desx, y, h)
```

Arguments

u	numerical vector of argument values,
desx	numerical vecror of design points,
y	numerical vecror of data values (corresponding to the specified design points $desx$),
h	numerical bandwidth value (scalar).

Details

Computing the LLE based on the Gaussian kernel for the specified vector of the argument values u and given vectors of design points $desx$ and the corresponding data values y.

Value

Numerical vector of the LLE values computed over the specified vector of u points.

References

- Clevelend, W.S. (1979). Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74(368), 829-836.
- Savchuk, O.Y., Hart, J.D. (2017). Fully robust one-sided cross-validation for regression functions. *Computational Statistics*, doi:10.1007/s00180-017-0713-7.

See Also

OSCV_reg, h_OSCV_reg, ASE_reg, h_ASE_reg, CV_reg.

Examples

```
## Not run:
# Example (simulated data).
n=200
dx=(1:n-0.5)/n
regf=2*dx^10*(1-dx)^2+dx^2*(1-dx)^10
u=seq(0,1,len=1000)
ydat=regf+rnorm(n,sd=0.002)
dev.new()
plot(dx,regf,'l',lty="dashed",lwd=3,xlim=c(0,1),ylim=c(1.1*min(ydat),1.1*max(ydat)),
cex.axis=1.7,cex.lab=1.7)
title(main="Function, generated data, and LLE",cex.main=1.5)
points(dx,ydat,pch=20,cex=1.5)
lines(u,loclin(u,dx,ydat,0.05),lwd=3,col="blue")
legend(0,1.1*max(ydat),legend=c("LLE based on h=0.05","true regression function"),
lwd=c(2,3),lty=c("solid","dashed"),col=c("blue","black"),cex=1.5,bty="n")
legend(0.7,0.5*min(ydat),legend="n=200",cex=1.7,bty="n")

## End(Not run)
```

L_I *The family of one-sided cross-validation kernels L_I.*

Description

The one-sided counterpart of the kernel H_I. See expressions (15) and (8) of Savchuk and Hart (2017).

Usage

```
L_I(u, alpha, sigma)
```

Arguments

u	numerical vector of argument values,
alpha	first parameter of the cross-validation kernel L_I,
sigma	second parameter of the cross-validation kernel L_I.

Details

The family of the one-sided cross-validation kernels L_I indexed by the parameters $-\infty < \alpha < \infty$ and $\sigma > 0$. This family is used in the OSCV implementations in both regression context (see Savchuk and Hart (2017)) and density estimation context (see Savchuk (2017)). The special members of the family:

- The *robust* kernel used in Savchuk and Hart (2017) and Savchuk (2017) is obtained by setting $\alpha = 16.8954588$ and $\sigma = 1.01$;
- The one-sided Gaussian kernel L_G is obtained by either setting $\alpha = 0$ for any $\sigma > 0$ or by setting $\sigma = 1$ for any $-\infty < \alpha < \infty$.

The bandwidth selected by L_I should be multiplied by a reascaling constant before it is used in computing the ultimate Gaussian (regression or density) estimate. In the case of a smooth (regression or density) function the rescaling constant is C_smooth.

Value

The value of $L_I(u; \alpha, \sigma)$.

References

- Savchuk, O.Y., Hart, J.D. (2017). Fully robust one-sided cross-validation for regression functions. *Computational Statistics*, doi:10.1007/s00180-017-0713-7.
- Savchuk, O.Y. (2017). One-sided cross-validation for nonsmooth density functions, arXiv:1703.05157.

See Also

H_I, C_smooth, OSCV_LI_dens.

Examples

```
## Not run:
# Plotting the robust one-sided kernel from Savchuk and Hart (2017) with
# alpha=16.8954588 and sigma=1.01.
u=seq(-1,5,len=1000)
rker=L_I(u,16.8954588,1.01)
Gker=L_I(u,0,1)
dev.new()
plot(u,rker,'l',lwd=3,cex.axis=1.7, cex.lab=1.7)
title(main="One-sided kernels: L_I (robust) and L_G",cex=1.7)
lines(u,Gker,lty="dashed",lwd=3)
legend(0.5,2.5,lty=c("solid","dashed"),lwd=c(3,3),legend=c("L_I","L_G"),cex=1.7)
legend(2,1.5,legend=c("alpha=16.8955","sigma=1.01"),cex=1.5)
```

```
## End(Not run)
```

`OSCV_Epan_dens` *The OSCV function based on L_E, the one-sided Epanechnikov kernel, in the kernel density estimation (KDE) context.*

Description

Computing the values of the L_E-based OSCV function in the density estimation context. See Martinez-Miranda et al. (2009) and Savchuk (2017).

Usage

```
OSCV_Epan_dens(h, dat)
```

Arguments

`h`	numerical vector of bandwidth values,
`dat`	numerical vecror of data values.

Details

Computing the values of the OSCV function for the given bandwidth vector h and data vector dat. The function is based on the one-sided Epanechnikov kernel L_E. The function's minimizer is to be multiplied by the appropriate rescaling constant before it can be used to compute the ultimate kernel density estimate. The formula for the rescaling constant depends on *smothness* of the density and on the *kernel* used in computing the ultimate density estimate.

Value

The vector of values of the OSCV function for the correponsing vector of h values.

References

- Martinez-Miranda, M.D., Nielsen, J. P., and Sperlich, S. (2009). One sided cross validation for density estimation. In *Operational Risk Towards Basel III: Best Practices and Issues in Modeling, Management and Regulation*, 177-196.
- Savchuk, O.Y. (2017). One-sided cross-validation for nonsmooth densty functions, arXiv:1703.05157.

See Also

`OSCV_Gauss_dens`, `OSCV_LI_dens`.

Examples

```
## Not run:
# Example 1 (Data on n=272 eruption duration of the Old Faithful geyser).
data=faithful[,1]
har=seq(0.05,1,len=1000)
dev.new()
plot(har,OSCV_Epan_dens(har,data),lwd=3,'l',xlab="h",ylab="L_E-based OSCV",
main="L_E_based OSCV for the data on eruption duration",cex.main=1.5,cex.lab=1.7,cex.axis=1.7)
h_min=round(optimize(OSCV_Epan_dens,c(0.001,1),tol=0.001,dat=data)$minimum, digits=4)
legend(0.1,-0.1,legend=c("n=272",paste("h_min=",h_min)),cex=2)
# The above graph appears in Savchuk (2017).

# Example 2 (Data set of size n=100 is generated from the standard normal density).
dat_norm=rnorm(100)
harray=seq(0.25,4.25,len=1000)
OSCVarray=OSCV_Epan_dens(harray,dat_norm)
dev.new()
plot(harray,OSCVarray,lwd=3,'l',xlab="h",ylab="L_E-based OSCV",
main="L_E-based OSCV for data generated from N(0,1)", cex.main=1.5,cex.lab=1.7,cex.axis=1.7)
h_min_norm=round(optimize(OSCV_Epan_dens,c(0.1,4),tol=0.001,dat=dat_norm)$minimum, digits=4)
legend(0.5,OSCVarray[1],legend=c("n=100",paste("h_min=",h_min_norm)),cex=2,bty="n")

## End(Not run)
```

`OSCV_Gauss_dens` *The OSCV function based on L_G, the one-sided Gaussian kernel, in the kernel density estimation (KDE) context.*

Description

Computing the values of the L_G-based OSCV function in the density estimation context. See Savchuk (2017).

Usage

```
OSCV_Gauss_dens(h, dat, stype)
```

Arguments

`h`	numerical vector of bandwidth values,
`dat`	numerical vecror of data values,
`stype`	specifies (anticipated) smoothness of the density function. Thus, $stype = 0$ corresponds to the *smooth* density, whereas $stype = 1$ corresponds to the *non-smooth* density.

Details

Computing the values of the OSCV function for the given bandwidth vector h and data vector dat. The function is based on the one-sided Gaussian kernel L_G. The (anticipated) smoothness of the underlying density function is to be specified. Thus,

- $stype = 0$ corresponds to the smooth density;
- $stype = 1$ corresponds to the nonsmooth density.

It is usually assumed that the density is smooth if no preliminary information about its nonsmoothness is available. The function's minimizer h_OSCV_dens is to be used without additional rescaling to compute the ultimate Gaussian density estimate.

Value

The vector of values of the OSCV function for the correponsing vector of h values.

References

Savchuk, O.Y. (2017). One-sided cross-validation for nonsmooth densty functions, arXiv:1703.05157.

See Also

h_OSCV_dens, OSCV_Epan_dens, OSCV_LI_dens, C_smooth.

Examples

```
## Not run:
dat_norm=rnorm(300)  #generating random sample of size n=300 from the standard normal density.
h_oscv=round(h_OSCV_dens(dat_norm,0),digits=4)
y=density(dat_norm,bw=h_oscv)
dev.new()
plot(y,lwd=3,cex.lab=1.7,cex.axis=1.7,cex.main=1.7,xlab=paste("n=100, h_OSCV=",h_oscv),
main="Standard normal density estimate by OSCV",ylim=c(0,0.45),xlim=c(-4.5,4.5))
u=seq(-5,5,len=1000)
lines(u,dnorm(u),lwd=3,lty="dashed",col="blue")
legend(0.75,0.4,legend=c("OSCV estimate","N(0,1) density"),lwd=c(3,3),lty=c("solid","dashed"),
col=c("black","blue"),bty="n",cex=1.25)

## End(Not run)
```

OSCV_LI_dens *The OSCV function based on the kernel L_I in the density estimation (KDE) context.*

Description

Computing the values of the L_I-based OSCV function in the density estimation context. See Savchuk (2017).

Usage

```
OSCV_LI_dens(h, dat, alpha, sigma)
```

Arguments

h	numerical vector of bandwidth values,
dat	numerical vecror of data values,
alpha	first parameter of the kernel L_I,
sigma	second parameter of the kernel L_I.

Details

Computing the OSCV function for the given vector of bandwidth values h and the data vector dat. The function is based on the one-sided kernel L_I that depends on the parameters α and σ. The kernel L_I is robust in the special case of $\alpha = 16.8954588$ and $\sigma = 1.01$. The other special case is obtained when either of the following holds:

- $\alpha = 0$ for any $\sigma > 0$;
- $\sigma = 1$ for any $-\infty < \alpha < \infty$.

In the above cases the kernel L_I reduces to the one-sided Gaussian kernel L_G. The function's minimizer is to be used without additional rescaling to compute the ultimate Gaussian density estimate under the assumption that the underlying density is smooth.

Value

The vector of values of the OSCV function for the correponsing vector of h values.

References

Savchuk, O.Y. (2017). One-sided cross-validation for nonsmooth density functions, arXiv:1703.05157.

See Also

OSCV_Gauss_dens, OSCV_Epan_dens, C_smooth, L_I, H_I.

Examples

```
## Not run:
# Example 1 (Old Faithful geyser data)
dev.new()
data=faithful[,1]         # Data on n=272 eruption duration of the Old Faithful geyser.
harray=seq(0.025,0.6,len=50)
alp=16.8954588
sig=1.01
plot(harray,OSCV_LI_dens(harray,data,alpha=alp,sigma=sig),lwd=3,'l',xlab="h",
ylab="L_I-based OSCV",main="OSCV_LI(h) for eruption duration",cex.main=1.5,cex.lab=1.7,
cex.axis=1.7)
h_OSCV_LI=round(optimize(OSCV_LI_dens,c(0.001,0.5),tol=0.001,dat=data,alpha=16.8954588,
sigma=1.01)$minimum,digits=4)
```

```
legend(0.01,-0.2,legend=c("n=272",paste("h_OSCV_LI=",h_OSCV_LI)),cex=1.8,bty="n")
legend(0.25,-0.33,legend=c("Parameters of L_I:", paste("alpha=",alp),
paste("sigma=",sig)),cex=1.7,bty="n")

# Example 2 (Simulated example)
dat_norm=rnorm(100)   #generating a random sample of size n=100 from the N(0,1) density
harray=seq(0.05,1.5,len=100)
OSCVarray=OSCV_LI_dens(harray,dat=dat_norm,16.8954588,1.01)
dev.new()
plot(harray,OSCVarray,lwd=3,'l',xlab="h",ylab="L_I-based OSCV",
main="OSCV_LI(h) for data generated from N(0,1)",cex.main=1.5,cex.lab=1.7,cex.axis=1.7)
h_OSCV_LI_norm=round(optimize(OSCV_LI_dens,c(0.001,1),tol=0.001,
dat=dat_norm,16.8954588,1.01)$minimum,digits=4)
legend(0,OSCVarray[1],legend=c("n=100",paste("h_OSCV_LI=",h_OSCV_LI_norm),
"Parameters of the robust kernel L_I:","alpha=16.8954588", "sigma=1.01"),cex=1.5,bty="n")

## End(Not run)
```

OSCV_reg *The OSCV function in the regression context.*

Description

Computing $OSCV(b)$, the value of the OSCV function in the regression context, defined by expression (9) of Savchuk and Hart (2017).

Usage

```
OSCV_reg(b, desx, y, ktype)
```

Arguments

b	numerical vector of bandwidth values,
desx	numerical vecror of design points,
y	numerical vecror of data points corresponding to the design points $desx$,
ktype	making choice between two cross-validation kernels: ($ktype = 0$) corresponds to the Gaussian kernel; ($ktype = 1$) corresponds to the robust kernel H_I with $(\alpha, \sigma) = (16.8954588, 1.01)$.

Details

Computation of $OSCV(b)$ for given b (bandwidth vector) and the data values y corresponding to the design points $desx$. No preliminary sorting of the data (according to the $desx$ variable) is needed. The value of $m = 4$ is used. Two choices of the two-sided cross-validation kernel are available:

- ($ktype = 0$) Gaussian kernel;
- ($ktype = 1$) robust kernel H_I defined by expression (15) of Savchuk and Hart (2017) with $(\alpha, \sigma) = (16.8954588, 1.01)$.

Value

The vector of values of $OSCV(b)$ for the correponsing vector of b values.

References

- Savchuk, O.Y., Hart, J.D. (2017). Fully robust one-sided cross-validation for regression functions. *Computational Statistics*, doi:10.1007/s00180-017-0713-7.
- Hart, J.D. and Yi, S. (1998) One-sided cross-validation. *Journal of the American Statistical Association*, 93(442), 620-631.

See Also

h_OSCV_reg, H_I, loclin, C_smooth.

Examples

```
## Not run:
# The Old Faithful geyser data set "faithful" is used. The sample size n=272.
# The OSCV curves based on the Gaussian kernel and the robust kernel H_I (with
# alpha=16.8954588 and sigma=1.01) are plotted. The horizontal scales of the curves
# are changed such that their global minimizers are to be used in computing the
# Gaussian local linear estimates of the regression function.
xdat=faithful[[2]] #waiting time
ydat=faithful[[1]] #eruption duration
barray=seq(0.5,10,len=250)
C_gauss=C_smooth(1,1)
OSCV_gauss=OSCV_reg(barray/C_gauss,xdat,ydat,0)
h_gauss=round(h_OSCV_reg(xdat,ydat,0),digits=4)
dev.new()
plot(barray,OSCV_gauss,'l',lwd=3,cex.lab=1.7,cex.axis=1.7,xlab="h",ylab="OSCV criterion")
title(main="OSCV based on the Gaussian kernel",cex.main=1.7)
legend(2.5,0.25,legend=paste("h_min=",h_gauss),cex=2,bty="n")
C_H_I=C_smooth(16.8954588,1.01)
OSCV_H_I=OSCV_reg(barray/C_H_I,xdat,ydat,1)
h_H_I=round(barray[which.min(OSCV_H_I)],digits=4)
dev.new()
plot(barray,OSCV_H_I,'l',lwd=3,cex.lab=1.7,cex.axis=1.7,xlab="h",ylab="OSCV criterion",
ylim=c(0.15,0.5))
title(main="OSCV based on the robust kernel H_I",cex.main=1.7)
legend(2.5,0.4,legend=paste("h_min=",h_H_I),cex=2,bty="n")

## End(Not run)
```

---

reg3 *Nonsmooth regression function with six cusps.*

Description

Nonsmooth regression function r_3 with six cusps used in the simulation studies in Savchuk et al. (2013) and Savchuk et al. (2017).

Usage

```
reg3(u)
```

Arguments

u	numerical vecror of argument values in the range [0,1].

Details

The nonsmooth function r_3 can be used in simulation studies.

Value

The vector of values of r_3 corresponding to the values of the vector u.

References

- Savchuk, O.Y., Hart, J.D., Sheather, S.J. (2013). One-sided cross-validation for nonsmooth regression functions. *Journal of Nonparametric Statistics*, 25(4), 889-904.
- Savchuk, O.Y., Hart, J.D. (2017). Fully robust one-sided cross-validation for regression functions. *Computational Statistics*, doi:10.1007/s00180-017-0713-7.

Examples

```
## Not run:
# n=250 data points are generated from r3 by adding the Gaussian noise with sigma=1/500.
# The fixed evenly spaced design is used.
u=seq(0,1,len=1000)
n=250
xdat=(1:n-0.5)/n
ydat=reg3(xdat)+rnorm(n,sd=1/500)
h_oscv=round(h_OSCV_reg(xdat,ydat,1),digits=4) # L_G-based OSCV based on nonsmooth constant
l=loclin(u,xdat,ydat,h_oscv)
dev.new()
plot(xdat,ydat,pch=20,cex=1.5,cex.axis=1.5,cex.lab=1.5,xlab="x",ylab="y",
ylim=c(min(ydat),1.2*max(ydat)))
lines(u,l,'l',lwd=3,col="blue")
lines(u,reg3(u),lwd=3,lty="dashed")
title(main="Data, true regression function and LLE",cex.main=1.7)
legend(-0.05,0.003,legend=paste("h_OSCV=",h_oscv),cex=2,bty="n")
legend(0.65,0.025, legend="n=250",cex=2,bty="n")
legend(0,1.28*max(ydat),legend=c("LLE based on h_OSCV","true regression function"),lwd=c(3,3),
lty=c("solid","dashed"),col=c("blue","black"),bty="n",cex=1.5)

## End(Not run)
```

sample_fstar *Taking a random sample from* fstar.

Description

Taking a random sample of size n from the density f^* with seven cusps introduced in the article of Savchuk (2017).

Usage

```
sample_fstar(n)
```

Arguments

n	sample size.

Details

The density f^* can be used in simulation studies.

Value

The numerical vector of size n of the data values.

References

Savchuk, O.Y. (2017). One-sided cross-validation for nonsmooth density functions, arXiv:1703.05157.

See Also

fstar, ISE_fstar.

Examples

```
## Not run:
dev.new()
plot(density(sample_fstar(5000),bw=0.1),lwd=2,ylim=c(0,0.32),xlab="argument",ylab="density",
main="KDE and the true density fstar",cex.lab=1.7, cex.axis=1.7,cex.main=1.7)
lines(seq(-3.5,3.5,len=1000),fstar(seq(-3.5,3.5,len=1000)),lwd=3,lty="dashed")
legend(-3,0.3,legend=c("KDE","True density","h=0.1","n=5000"),lwd=c(2,3),
lty=c("solid","dashed"),col=c("black","black","white","white"))

## End(Not run)
```

Index