Package 'cvmgof'

December 3, 2018

Title Cramer-von Mises Goodness-of-Fit Tests

Type Package

Version 1.0.0

Author Romain Azais, Sandie Ferrigno and Marie-Jose Martinez

Maintainer Romain Azais <romain.azais@inria.fr>

Description It is devoted to Cramer-von Mises goodness-of-fit tests.
It implements three statistical methods based on Cramer-
von Mises statistics to estimate and test a regression model.

License CeCILL

Encoding UTF-8

LazyData FALSE

Imports lattice

RoxygenNote 6.1.0

NeedsCompilation no

Repository CRAN

Date/Publication 2018-12-03 19:50:09 UTC

R topics documented:

acgm.bandwidth.selection.linkfunction

Bandwidth selection of the link function under the null hypothesis

Description

This function computes the optimal bandwidth of the link function under the null hypothesis

Usage

```
acgm.bandwidth.selection.linkfunction(data.X.H0, data.Y.H0, linkfunction.H0,
kernel.function = kernel.function.epan, verbose = TRUE)
```

Arguments

data.X.H0   a numeric data vector used to obtain the nonparametric estimator of the regression function under the null hypothesis.

data.Y.H0   a numeric data vector used to obtain the nonparametric estimator of the regression function under the null hypothesis.

linkfunction.H0
                   regression function under the null hypothesis

kernel.function
                   kernel function used to obtain the nonparametric estimator of the regression function. Default option is "kernel.function.epan" which corresponds to the Epanechnikov kernel function.

verbose   If TRUE, the R function plots the link function (regression function) under the null hypothesis and the local linear link function estimation on a same graph. Default option is TRUE.

Author(s)

Romain Azais, Sandie Ferrigno and Marie-Jose Martinez

References

J. T. Alcala, J. A. Cristobal, and W. Gonzalez Manteiga. Goodness-of-fit test for linear models based on local polynomials. *Statistics & Probability Letters*, 42(1), 39:46, 1999.

R. Azais, S. Ferrigno and M-J Martinez. cvmgof: An R package for Cramér-von Mises goodness-of-fit tests in regression models. 2018. Preprint in progress.

Examples

```
set.seed(1)

# Data simulation
n = 25 # Dataset size
data.X = runif(n,min=0,max=5) # X
data.Y = 0.2*data.X^2-data.X+2+rnorm(n,mean=0,sd=0.3) # Y

########################################################################

# Bandwidth selection under H0

# We generate a dataset under H0 to estimate the optimal bandwidth under H0

linkfunction.H0 = function(x){0.2*x^2-x+2}

data.X.H0 = runif(n,min=0,max=5)
data.Y.H0 = linkfunction.H0(data.X.H0)+rnorm(n,mean=0,sd=0.3)

h.opt.acgm = acgm.bandwidth.selection.linkfunction(data.X.H0 , data.Y.H0,linkfunction.H0)
```

---

acgm.linkfunction.estim

Local linear estimation of the regression function

Description

This function computes the local linear estimation of the regression function.

Usage

```
acgm.linkfunction.estim(x, data.X, data.Y, bandwidth,
kernel.function = kernel.function.epan)
```

Arguments

x	a numeric vector.
data.X	a numeric data vector used to obtain the nonparametric estimator of the regression function.
data.Y	a numeric data vector used to obtain the nonparametric estimator of the regression function.
bandwidth	bandwidth used to obtain the nonparametric estimator of the regression function.
kernel.function	
	kernel function used to obtain the nonparametric estimator of the regression function. Default option is "kernel.function.epan" which corresponds to the Epanechnikov kernel function.

Details

Inappropriate bandwidth or x choices can produce "NaN" values in link function estimates.

Author(s)

Romain Azais, Sandie Ferrigno and Marie-Jose Martinez

References

J. T. Alcala, J. A. Cristobal, and W. Gonzalez Manteiga. Goodness-of-fit test for linear models based on local polynomials. *Statistics & Probability Letters*, 42(1), 39:46, 1999.

R. Azais, S. Ferrigno and M-J Martinez. cvmgof: An R package for Cramér-von Mises goodness-of-fit tests in regression models. 2018. Preprint in progress.

Examples

```
set.seed(1)

# Data simulation
n = 25 # Dataset size
data.X = runif(n,min=0,max=5) # X
data.Y = 0.2*data.X^2-data.X+2+rnorm(n,mean=0,sd=0.3) # Y

########################################################################

# Estimation of the link function

bandwidth = 0.75 # Here, the bandwidth is arbitrarily fixed

xgrid = seq(0,5,by=0.1)
ygrid_acgm = acgm.linkfunction.estim(xgrid,data.X,data.Y,bandwidth)

plot(xgrid,ygrid_acgm,type='l',lty=1,lwd=2,xlab='X',ylab='Y',ylim=c(0.25,2.5))
lines(xgrid,0.2*xgrid^2-xgrid+2,lwd=0.5,col='gray')
```

---

acgm.statistics	*Local test statistic for the regression function*

Description

This function computes the local test statistic for the regression function.

Usage

```
acgm.statistics(data.X, data.Y, linkfunction.H0,
bandwidth = "optimal",
kernel.function = kernel.function.epan,
integration.step = 0.01,
verbose = TRUE)
```

Arguments

`data.X`	a numeric data vector used to obtain the nonparametric estimator of the regression function.
`data.Y`	a numeric data vector used to obtain the nonparametric estimator of the regression function.
`linkfunction.H0`	
	the regression function under the null hypothesis.
`bandwidth`	bandwidth used to obtain the nonparametric estimator of the regression function. If bandwidth = "optimal", the optimal bandwidth of the regression function under the null hypothesis is computed. Default option is "optimal".
`kernel.function`	
	kernel function used to obtain the nonparametric estimator of theregression function. Default option is "kernel.function.epan".
`integration.step`	
	a numeric value specifying integration step. Default is `integration.step =` 0.01.
`verbose`	If `TRUE`, the R function displays the optimal bandwidth value obtained under the null hypothesis. Default option is `TRUE`.

Author(s)

Romain Azais, Sandie Ferrigno and Marie-Jose Martinez

References

J. T. Alcala, J. A. Cristobal, and W. Gonzalez Manteiga. Goodness-of-fit test for linear models based on local polynomials. *Statistics & Probability Letters*, 42(1), 39:46, 1999.

R. Azais, S. Ferrigno and M-J Martinez. cvmgof: An R package for Cramér-von Mises goodness-of-fit tests in regression models. 2018. Preprint in progress.

Examples

```
set.seed(1)

# Data simulation
n = 25 # Dataset size
data.X = runif(n,min=0,max=5) # X
data.Y = 0.2*data.X^2-data.X+2+rnorm(n,mean=0,sd=0.3) # Y

########################################################################

# Bandwidth selection under H0

# We want to test if the link function is f(x)=0.2*x^2-x+2
# The answer is yes (see the definition of data.Y above)
# We generate a dataset under H0 to estimate the optimal bandwidth under H0

linkfunction.H0 = function(x){0.2*x^2-x+2}
```

```
data.X.H0 = runif(n,min=0,max=5)
data.Y.H0 = linkfunction.H0(data.X.H0)+rnorm(n,mean=0,sd=0.3)

h.opt.acgm = acgm.bandwidth.selection.linkfunction(data.X.H0 , data.Y.H0,linkfunction.H0)

##########################################################################

# Test statistics under H0

acgm.statistics(data.X,data.Y,linkfunction.H0,h.opt.acgm)
```

---

acgm.test.bootstrap   *Local test for the regression function*

Description

A local test for the regression function.

Usage

```
acgm.test.bootstrap(data.X, data.Y, linkfunction.H0, risk,
 bandwidth = "optimal",
   kernel.function = kernel.function.epan,
   bootstrap = c(50, "Mammen"),
   integration.step = 0.01,
   verbose=TRUE)
```

Arguments

data.X	a numeric data vector used to obtain the nonparametric estimator of the regression function.
data.Y	a numeric data vector used to obtain the nonparametric estimator of the regression function.
linkfunction.H0	
	the regression function under the null hypothesis.
risk	a numeric value specifying the risk of rejecting the null hypothesis. The value (1-risk) corresponds to the confidence level of the statistical test.
bandwidth	the bandwidth used to obtain the nonparametric estimator of the regression function. If bandwidth = "optimal", the optimal bandwidth of the regression function under the null hypothesis is computed. Default option is "optimal".
kernel.function	
	the kernel function used to obtain the nonparametric estimator of the regression function. Default option is "kernel.function.epan".

bootstrap	a numeric vector of length 2. The first value specifies the number of bootstrap datasets (default is "50"). The second value specifies the distribution used for the wild bootstrap resampling (default is "Mammen").
integration.step	
	a numeric value specifying integration step. Default is integration.step = 0.01.
verbose	If TRUE, the R function displays the optimal bandwidth value obtained under the null hypothesis. Default option is TRUE.

Details

From data.X and data.Y datasets, wild bootstrap datasets ("50" by default) are built. From each bootstrap dataset, a bootstrap test statistic is computed. The test statistic under the null hypothesis is compared to the distribution of the bootstrap statistics. The test is rejected if the test statistic under the null hypothesis is greater than the (1-risk)-quantile of the empirical distribution of the bootstrap statistics.

An inappropriate bandwidth choice can produce "NaN" values in test statistics.

Value

acgm.test.bootstrap returns a list containing the following components:

decision	the statistical decision made on whether to reject the null hypothesis or not.
bandwidth	the bandwidth used to build the statistics test.
pvalue	the p-value of the test statistics.
test_statistics	
	the test statistics value.

Author(s)

Romain Azais, Sandie Ferrigno and Marie-Jose Martinez

References

J. T. Alcala, J. A. Cristobal, and W. Gonzalez Manteiga. Goodness-of-fit test for linear models based on local polynomials. *Statistics & Probability Letters*, 42(1), 39:46, 1999.

R. Azais, S. Ferrigno and M-J Martinez. cvmgof: An R package for Cramér-von Mises goodness-of-fit tests in regression models. 2018. Preprint in progress.

Examples

```
set.seed(1)

# Data simulation
n = 25 # Dataset size
data.X = runif(n,min=0,max=5) # X
data.Y = 0.2*data.X^2-data.X+2+rnorm(n,mean=0,sd=0.3) # Y
```

```
##########################################################################

# Test (bootstrap) under H0

# We want to test if the link function is f(x)=0.2*x^2-x+2
# The answer is yes (see the definition of data.Y above)
# We generate a dataset under H0 to estimate the optimal bandwidth under H0

linkfunction.H0 = function(x){0.2*x^2-x+2}

test_acgm.H0 = acgm.test.bootstrap(data.X,data.Y,linkfunction.H0,
0.05,bandwidth='optimal',bootstrap=c(50,'Mammen'),integration.step = 0.01)


##########################################################################

# Test (bootstrap) under H1

# We want to test if the link function is f(x)=0.5*cos(x)+1
# The answer is no (see the definition of data.Y above)

linkfunction.H1=function(x){0.8*cos(x)+1}

test_acgm.H1 = acgm.test.bootstrap(data.X,data.Y,linkfunction.H1,0.05,
bandwidth='optimal',bootstrap=c(50,'Mammen'),integration.step = 0.01)
```

---

cvmgof   *cvmgof*

Description

It implements three goodness-of-fit tests to test the validity of the regression function in the regression model. Two of them (Alcala *et al.* '99 and Van Keilegom *et al.* '12) are "directional" in that they detect departures from mainly the regression function assumption of the model or "global" (Ducharme and Ferrigno '12) with the conditional distribution function. The establishment of such statistical tests requires nonparametric estimators and the use of wild bootstrap methods for the simulations.

Details

Package:	cvmgof
Type:	Package
Version:	1.0.0
Date:	2018-11-29
License:	Cecill

Author(s)

Romain Azais, Sandie Ferrigno and Marie-Jose Martinez

References

J. T. Alcala, J. A. Cristobal, and W. Gonzalez Manteiga. Goodness-of-fit test for linear models based on local polynomials. *Statistics & Probability Letters*, 42(1), 39:46, 1999.

G. R. Ducharme and S. Ferrigno. An omnibus test of goodness-of-fit for conditional distributions with applications to regression models. *Journal of Statistical Planning and Inference*, 142, 2748:2761, 2012.

I. Van Keilegom, W. Gonzalez Manteiga, and C. Sanchez Sellero. Goodness-of-fit tests in parametric regression based on the estimation of the error distribution. *Test*, 17, 401:415, 2008.

R. Azais, S. Ferrigno and M-J Martinez. cvmgof: An R package for Cramér-von Mises goodness-of-fit tests in regression models. 2018. Preprint in progress.

See Also

acgm.test.bootstrap

df.test.bootstrap

vkgmss.test.bootstrap

Examples

```
set.seed(1)
# The following example tests are computed from only 25 data points
# The seed is fixed to avoid NA in estimates

require(lattice) # Only for plotting conditional CDF

# Data simulation
n = 25 # Dataset size
data.X = runif(n,min=0,max=5) # X
data.Y = 0.2*data.X^2-data.X+2+rnorm(n,mean=0,sd=0.3) # Y

plot(data.X,data.Y,xlab='X',ylab='Y',pch='+')

#########################################################################

# Estimation of the link function

bandwidth = 0.75 # Here, the bandwidth is arbitrarily fixed

xgrid = seq(0,5,by=0.1)
ygrid_df = df.linkfunction.estim(xgrid,data.X,data.Y,bandwidth)
ygrid_acgm = acgm.linkfunction.estim(xgrid,data.X,data.Y,bandwidth)
ygrid_vkgmss = vkgmss.linkfunction.estim(xgrid,data.X,data.Y,bandwidth)
```

```
plot(xgrid,ygrid_df,type='l',col='blue',lty=1,lwd=2,xlab='X',ylab='Y',ylim=c(0.25,2.5))
lines(xgrid,ygrid_acgm,type='l',col='red',lty=2,lwd=2)
lines(xgrid,ygrid_vkgmss,type='l',col='dark green',lty=3,lwd=2)
lines(xgrid,0.2*xgrid^2-xgrid+2,lwd=0.5,col='gray')
# Ducharme and Ferrigno: blue
# Alcala et al.: red
# Van Keilegom et al.: dark green
# true link function: gray

# Estimation of the conditional CDF (only Ducharme and Ferrigno estimator)
xgrid = seq(0.5,4.5,by=0.1)
ygrid = seq(-1,3,by=0.1)
cdf_df = df.cdf.estim(xgrid,ygrid,data.X,data.Y,bandwidth)

wireframe(cdf_df, drape=TRUE,
    col.regions=rainbow(100),
    zlab='CDF(y|x)',
    xlab='x',ylab='y',zlim=c(0,1.01))

# Estimation of residuals cdf (only Van Keilegom et al. estimator)

egrid = seq(-5,5,by=0.1)
res.cdf_vkgmss = vkgmss.residuals.cdf.estim(egrid,data.X,data.Y,0.5)

plot(egrid,res.cdf_vkgmss,type='l',xlab='e',ylab='CDF(e)')

# Estimation of residuals standard deviation (only Van Keilegom et al. estimator)

sd_vkgmss = vkgmss.sd.estim(xgrid,data.X,data.Y,bandwidth)

plot(xgrid,sd_vkgmss, type='l',xlab='X',ylab='SD(X)')
abline(h=0.3)

########################################################################

# Bandwidth selection under H0

# We want to test if the link function is f(x)=0.2*x^2-x+2
# The answer is yes (see the definition of data.Y above)
# We generate a dataset under H0 to estimate the optimal bandwidth under H0

linkfunction.H0 = function(x){0.2*x^2-x+2}

data.X.H0 = runif(n,min=0,max=5)
data.Y.H0 = linkfunction.H0(data.X.H0)+rnorm(n,mean=0,sd=0.3)

h.opt.df = df.bandwidth.selection.linkfunction(data.X.H0 , data.Y.H0,linkfunction.H0)
h.opt.acgm = acgm.bandwidth.selection.linkfunction(data.X.H0 , data.Y.H0,linkfunction.H0)
h.opt.vkgmss = vkgmss.bandwidth.selection.linkfunction(data.X.H0 , data.Y.H0,linkfunction.H0)

########################################################################
```

```
# Test statistics under H0

# Remainder:
# Ducharme and Ferrigno test is on the conditional CDF and not on the link function
# Thus we need to define the conditional CDF associated
# with the link function under H0 to evaluate this test
# Alcala et al. and Van Keilegom et al. tests are on the link function

cond_cdf.H0 = function(x,y)
{
  out=matrix(0,nrow=length(x),ncol=length(y))
  for (i in 1:length(x)){
    x0=x[i]
    out[i,]=pnorm(y-linkfunction.H0(x0),0,0.3)
  }
  out
}
# cond_cdf.H0 is the conditional CDF associated with linkfunction.H0
# with additive Gaussian noise (standard deviation=0.3)

df.statistics(data.X,data.Y,cond_cdf.H0,h.opt.df)
acgm.statistics(data.X,data.Y,linkfunction.H0,h.opt.acgm)
vkgmss.statistics(data.X,data.Y,linkfunction.H0,h.opt.vkgmss)

########################################################################

# Test (bootstrap) under H0

test_df.H0 = df.test.bootstrap(data.X,data.Y,cond_cdf.H0,
  0.05,h.opt.df,bootstrap=c(50,'Mammen'),integration.step = 0.01)
test_acgm.H0 = acgm.test.bootstrap(data.X,data.Y,linkfunction.H0,
  0.05,bandwidth='optimal',bootstrap=c(50,'Mammen'),integration.step = 0.01)
test_vkgmss.H0 = vkgmss.test.bootstrap(data.X,data.Y,linkfunction.H0,
  0.05,bandwidth='optimal',bootstrap=c(50,'Mammen'))

# Example:
# test_acgm$decision is a string: 'accept H0' or 'reject H0'
# test_acgm$bandwidth is a float (optimal bandwidth under H0
# (only for Alcala and Van Keilegom tests) if bandwidth = 'optimal')
# test_acgm$pvalue is a float but it could be a string
# ('< 0.02' for instance or 'None' if the test can not be evaluated)
# test_acgm$test_statistics is a float but it could be a string
# ('None' if the test can not be evaluated)

# The 3 tests accept H0

########################################################################

# Test (bootstrap) under H1

# We want to test if the link function is f(x)=0.5*cos(x)+1
# The answer is no (see the definition of data.Y above)
```

```
linkfunction.H1=function(x){0.8*cos(x)+1}

plot(xgrid,linkfunction.H0(xgrid),type='l',ylim=c(-1,2))
lines(xgrid,linkfunction.H1(xgrid),type='l')

data.X.H1 = data.X.H0
data.Y.H1 = linkfunction.H1(data.X.H1)+rnorm(n,mean=0,sd=0.3)
h.opt.df = df.bandwidth.selection.linkfunction(data.X.H1 , data.Y.H1,linkfunction.H1)

cond_cdf.H1=function(x,y)
{
  out=matrix(0,nrow=length(x),ncol=length(y))
  for (i in 1:length(x)){
    x0=x[i]
    out[i,]=pnorm(y-linkfunction.H1(x0),0,0.3)
  }
  out
}

test_df.H1 = df.test.bootstrap(data.X,data.Y,cond_cdf.H1,
  0.05,h.opt.df,bootstrap=c(50,'Mammen'),integration.step = 0.01)
test_acgm.H1 = acgm.test.bootstrap(data.X,data.Y,linkfunction.H1,
  0.05,bandwidth='optimal',bootstrap=c(50,'Mammen'),integration.step = 0.01)
test_vkgmss.H1 = vkgmss.test.bootstrap(data.X,data.Y,linkfunction.H1,
  0.05,bandwidth='optimal',bootstrap=c(50,'Mammen'))

# From only 25 points, only Van Keilegom et al. test rejects H0
# while Ducharme and Ferrigno and Alcala et al. tests accept H0
```

---

df.bandwidth.selection.linkfunction

Bandwidth selection of the link function under the null hypothesis

---

Description

This function computes the optimal bandwidth of the link function under the null hypothesis.

Usage

```
df.bandwidth.selection.linkfunction(data.X.H0, data.Y.H0, linkfunction.H0,
kernel.function = kernel.function.epan, verbose = TRUE)
```

Arguments

data.X.H0   a numeric data vector used to obtain the nonparametric estimator of the condi-
                 tional distribution function under the null hypothesis.

data.Y.H0   a numeric data vector used to obtain the nonparametric estimator of the condi-
                 tional distribution function under the null hypothesis.

linkfunction.H0
> regression function under the null hypothesis

kernel.function
> kernel function used to obtain the nonparametric estimator of the conditional distribution function. Default option is "kernel.function.epan" which corresponds to the Epanechnikov kernel function.

verbose
> If TRUE, the R function plots the link function (regression function) under the null hypothesis and the local linear link function estimation on a same graph. Default option is TRUE.

Author(s)

Romain Azais, Sandie Ferrigno and Marie-Jose Martinez

References

G. R. Ducharme and S. Ferrigno. An omnibus test of goodness-of-fit for conditional distributions with applications to regression models. *Journal of Statistical Planning and Inference*, 142, 2748:2761, 2012.

R. Azais, S. Ferrigno and M-J Martinez. cvmgof: An R package for Cramér-von Mises goodness-of-fit tests in regression models. 2018. Preprint in progress.

Examples

```
set.seed(1)

# Data simulation
n = 25 # Dataset size
data.X = runif(n,min=0,max=5) # X
data.Y = 0.2*data.X^2-data.X+2+rnorm(n,mean=0,sd=0.3) # Y

#########################################################################

# Bandwidth selection under H0

# We generate a dataset under H0 to estimate the optimal bandwidth under H0

linkfunction.H0 = function(x){0.2*x^2-x+2}

data.X.H0 = runif(n,min=0,max=5)
data.Y.H0 = linkfunction.H0(data.X.H0)+rnorm(n,mean=0,sd=0.3)

h.opt.df = df.bandwidth.selection.linkfunction(data.X.H0 , data.Y.H0,linkfunction.H0)
```

df.cdf.estim   *Local linear estimation of the conditional distribution function*

Description

This function computes the local linear estimation of the conditional distribution function.

Usage

```
df.cdf.estim(x, y, data.X, data.Y, bandwidth, kernel.function = kernel.function.epan)
```

Arguments

x	a numeric vector.
y	a numeric vector.
data.X	a numeric data vector used to obtain the nonparametric estimator of the conditional distribution function.
data.Y	a numeric data vector used to obtain the nonparametric estimator of the conditional distribution function.
bandwidth	bandwidth used to obtain the nonparametric estimator of the conditional distribution function.
kernel.function	
	kernel function used to obtain the nonparametric estimator of the conditional distribution function. Default option is "kernel.function.epan" which corresponds to the Epanechnikov kernel function.

Details

Inappropriate bandwidth, x or y choices can produce "NaN" values in cumulative distribution function estimates.

Author(s)

Romain Azais, Sandie Ferrigno and Marie-Jose Martinez

References

G. R. Ducharme and S. Ferrigno. An omnibus test of goodness-of-fit for conditional distributions with applications to regression models. *Journal of Statistical Planning and Inference*, 142, 2748:2761, 2012.

R. Azais, S. Ferrigno and M-J Martinez. cvmgof: An R package for Cramér-von Mises goodness-of-fit tests in regression models. 2018. Preprint in progress.

Examples

```
set.seed(1)

require(lattice) # Only for plotting conditional CDF

# Data simulation
n = 25 # Dataset size
data.X = runif(n,min=0,max=5) # X
data.Y = 0.2*data.X^2-data.X+2+rnorm(n,mean=0,sd=0.3) # Y

########################################################################

# Estimation of the link function

bandwidth = 0.75 # Here, the bandwidth is arbitrarily fixed

# Estimation of the conditional CDF
xgrid = seq(0.5,4.5,by=0.1)
ygrid = seq(-1,3,by=0.1)
cdf_df = df.cdf.estim(xgrid,ygrid,data.X,data.Y,bandwidth)

wireframe(cdf_df, drape=TRUE,
col.regions=rainbow(100),zlab='CDF(y|x)',xlab='x',ylab='y',zlim=c(0,1.01))
```

---

df.linkfunction.estim *Local linear estimation of the regression function*

Description

This function computes the local linear estimation of the regression function using the local linear estimation of the conditional distribution function.

Usage

```
df.linkfunction.estim(x, data.X, data.Y, bandwidth,
kernel.function = kernel.function.epan)
```

Arguments

x	a numeric vector.
data.X	a numeric data vector used to obtain the nonparametric estimator of the conditional distribution function.
data.Y	a numeric data vector used to obtain the nonparametric estimator of the conditional distribution function.
bandwidth	bandwidth used to obtain the nonparametric estimator of the conditional distribution function.

kernel.function

> kernel function used to obtain the nonparametric estimator of the conditional distribution function. Default option is "kernel.function.epan" which corresponds to the Epanechnikov kernel function.

Details

Inappropriate bandwidth or x choices can produce "NaN" values in link function estimates.

Author(s)

Romain Azais, Sandie Ferrigno and Marie-Jose Martinez

References

G. R. Ducharme and S. Ferrigno. An omnibus test of goodness-of-fit for conditional distributions with applications to regression models. *Journal of Statistical Planning and Inference*, 142, 2748:2761, 2012.

R. Azais, S. Ferrigno and M-J Martinez. cvmgof: An R package for Cramér-von Mises goodness-of-fit tests in regression models. 2018. Preprint in progress.

Examples

```
set.seed(1)

# Data simulation
n = 25 # Dataset size
data.X = runif(n,min=0,max=5) # X
data.Y = 0.2*data.X^2-data.X+2+rnorm(n,mean=0,sd=0.3) # Y

#########################################################################

# Estimation of the link function

bandwidth = 0.75 # Here, the bandwidth is arbitrarily fixed

xgrid = seq(0,5,by=0.1)
ygrid_df = df.linkfunction.estim(xgrid,data.X,data.Y,bandwidth)

plot(xgrid,ygrid_df,type='l',lty=1,lwd=2,xlab='X',ylab='Y',ylim=c(0.25,2.5))
lines(xgrid,0.2*xgrid^2-xgrid+2,lwd=0.5,col='gray')
```

---

df.statistics   *Global test statistic for the conditional distribution function*

Description

This function computes the global test statistic for the conditional distribution function.

Usage

```
df.statistics(data.X, data.Y, cdf.H0, bandwidth,
    kernel.function = kernel.function.epan, integration.step = 0.01)
```

Arguments

data.X	a numeric data vector used to obtain the nonparametric estimator of the conditional distribution function.
data.Y	a numeric data vector used to obtain the nonparametric estimator of the conditional distribution function.
cdf.H0	the conditional distribution function under the null hypothesis.
bandwidth	bandwidth used to obtain the nonparametric estimator of the conditional distribution function.
kernel.function	
	kernel function used to obtain the nonparametric estimator of the conditional distribution function. Default option is "kernel.function.epan".
integration.step	
	a numeric value specifying integration step. Default is `integration.step = 0.01`.

Details

An inappropriate bandwidth choice can produce "NaN" values in cumulative distribution function estimates.

Author(s)

Romain Azais, Sandie Ferrigno and Marie-Jose Martinez

References

G. R. Ducharme and S. Ferrigno. An omnibus test of goodness-of-fit for conditional distributions with applications to regression models. *Journal of Statistical Planning and Inference*, 142, 2748:2761, 2012.

R. Azais, S. Ferrigno and M-J Martinez. cvmgof: An R package for Cramér-von Mises goodness-of-fit tests in regression models. 2018. Preprint in progress.

Examples

```
set.seed(1)

# Data simulation
n = 25 # Dataset size
data.X = runif(n,min=0,max=5) # X
data.Y = 0.2*data.X^2-data.X+2+rnorm(n,mean=0,sd=0.3) # Y

######################################################################
```

```
# Bandwidth selection under H0

# We want to test if the link function is f(x)=0.2*x^2-x+2
# The answer is yes (see the definition of data.Y above)
# We generate a dataset under H0 to estimate the optimal bandwidth under H0

linkfunction.H0 = function(x){0.2*x^2-x+2}

data.X.H0 = runif(n,min=0,max=5)
data.Y.H0 = linkfunction.H0(data.X.H0)+rnorm(n,mean=0,sd=0.3)

h.opt.df = df.bandwidth.selection.linkfunction(data.X.H0 , data.Y.H0,linkfunction.H0)

##########################################################################

# Test statistics under H0

cond_cdf.H0 = function(x,y)
{
  out=matrix(0,nrow=length(x),ncol=length(y))
  for (i in 1:length(x)){
    x0=x[i]
    out[i,]=pnorm(y-linkfunction.H0(x0),0,0.3)
  }
  out
}
# cond_cdf.H0 is the conditional CDF associated with linkfunction.H0
# with additive Gaussian noise (standard deviation=0.3)

df.statistics(data.X,data.Y,cond_cdf.H0,h.opt.df)
```

---

df.test.bootstrap   *Global test for the conditional distribution function*

Description

A global test for the conditional distribution function.

Usage

```
df.test.bootstrap(data.X, data.Y, cdf.H0, risk, bandwidth,
    kernel.function = kernel.function.epan,
    bootstrap = c(50, "Mammen"),
    integration.step = 0.01)
```

Arguments

data.X   a numeric data vector used to obtain the nonparametric estimator of the condi-
                    tional distribution function.

data.Y	a numeric data vector used to obtain the nonparametric estimator of the conditional distribution function.
cdf.H0	the conditional distribution function under the null hypothesis.
risk	a numeric value specifying the risk of rejecting the null hypothesis. The value (1-risk) corresponds to the confidence level of the statistical test.
bandwidth	the bandwidth used to obtain the nonparametric estimator of the conditional distribution function.
kernel.function	
	the kernel function used to obtain the nonparametric estimator of the conditional distribution function. Default option is "kernel.function.epan".
bootstrap	a numeric vector of length 2. The first value specifies the number of bootstrap datasets (default is "50"). The second value specifies the distribution used for the wild bootstrap resampling (default is "Mammen").
integration.step	
	a numeric value specifying integration step. Default is integration.step = 0.01.

Details

From data.X and data.Y datasets, wild bootstrap datasets ("50" by default) are built. From each bootstrap dataset, a bootstrap test statistic is computed. The test statistic under the null hypothesis is compared to the distribution of the bootstrap statistics. The test is rejected if the test statistic under the null hypothesis is greater than the (1-risk)-quantile of the empirical distribution of the bootstrap statistics.

An inappropriate bandwidth choice can produce "NaN" values in test statistics.

Value

df.test.bootstrap returns a list containing the following components:

decision	the statistical decision made on whether to reject the null hypothesis or not.
bandwidth	the bandwidth used to build the statistics test.
pvalue	the p-value of the test statistics.
test_statistics	
	the test statistics value.

Author(s)

Romain Azais, Sandie Ferrigno and Marie-Jose Martinez

References

G. R. Ducharme and S. Ferrigno. An omnibus test of goodness-of-fit for conditional distributions with applications to regression models. *Journal of Statistical Planning and Inference*, 142, 2748:2761, 2012.

R. Azais, S. Ferrigno and M-J Martinez. cvmgof: An R package for Cramér-von Mises goodness-of-fit tests in regression models. 2018. Preprint in progress.

Examples

```
set.seed(1)

# Data simulation
n = 25 # Dataset size
data.X = runif(n,min=0,max=5) # X
data.Y = 0.2*data.X^2-data.X+2+rnorm(n,mean=0,sd=0.3) # Y


##########################################################################

# Bandwidth selection under H0

# We want to test if the link function is f(x)=0.2*x^2-x+2
# The answer is yes (see the definition of data.Y above)
# We generate a dataset under H0 to estimate the optimal bandwidth under H0

linkfunction.H0 = function(x){0.2*x^2-x+2}

data.X.H0 = runif(n,min=0,max=5)
data.Y.H0 = linkfunction.H0(data.X.H0)+rnorm(n,mean=0,sd=0.3)

h.opt.df = df.bandwidth.selection.linkfunction(data.X.H0 , data.Y.H0,linkfunction.H0)

##########################################################################

# Test (bootstrap) under H0

# Remainder:
# Ducharme and Ferrigno test is on the conditional CDF and not on the link function
# Thus we need to define the conditional CDF associated
# with the link function under H0 to evaluate this test

cond_cdf.H0 = function(x,y)
{
  out=matrix(0,nrow=length(x),ncol=length(y))
  for (i in 1:length(x)){
    x0=x[i]
    out[i,]=pnorm(y-linkfunction.H0(x0),0,0.3)
  }
  out
}
# cond_cdf.H0 is the conditional CDF associated with linkfunction.H0
# with additive Gaussian noise (standard deviation=0.3)

# Test (bootstrap) under H0

test_df.H0 = df.test.bootstrap(data.X,data.Y,cond_cdf.H0,
  0.05,h.opt.df,bootstrap=c(50,'Mammen'),integration.step = 0.01)

##########################################################################
```

```
# Test (bootstrap) under H1

# We want to test if the link function is f(x)=0.5*cos(x)+1
# The answer is no (see the definition of data.Y above)

linkfunction.H1=function(x){0.8*cos(x)+1}

data.X.H1 = data.X.H0
data.Y.H1 = linkfunction.H1(data.X.H1)+rnorm(n,mean=0,sd=0.3)
h.opt.df = df.bandwidth.selection.linkfunction(data.X.H1 , data.Y.H1,linkfunction.H1)

cond_cdf.H1=function(x,y)
{
  out=matrix(0,nrow=length(x),ncol=length(y))
  for (i in 1:length(x)){
    x0=x[i]
    out[i,]=pnorm(y-linkfunction.H1(x0),0,0.3)
  }
  out
}

test_df.H1 = df.test.bootstrap(data.X,data.Y,cond_cdf.H1,
  0.05,h.opt.df,bootstrap=c(50,'Mammen'),integration.step = 0.01)
```

---

kernel.function.epan   *Epanechnikov kernel*

Description

This function computes the Epanechnikov kernel used to estimate the conditional distribution function.

Usage

```
kernel.function.epan(u)
```

Arguments

u   a numeric vector.

Author(s)

Romain Azais, Sandie Ferrigno and Marie-Jose Martinez

References

J. Fan and I. Gijbels. Local polynomial modelling and its applications. *Chapman & Hall*, London, 1996.

Examples

```
x = runif(10)  #simulating a uniform random sample
kernel.function.epan(x)
```

---

kernel.function.gauss *Gaussian kernel*

Description

This function computes the Gaussian kernel used to estimate the conditional distribution function.

Usage

```
kernel.function.gauss(u)
```

Arguments

u   a numeric vector.

Author(s)

Romain Azais, Sandie Ferrigno and Marie-Jose Martinez

References

J. Fan and I. Gijbels. Local polynomial modelling and its applications. *Chapman & Hall*, London, 1996.

Examples

```
x = runif(10)  #simulating a uniform random sample
kernel.function.gauss(x)
```

---

kernel.function.quart *Quartic kernel*

Description

This function computes the Quartic kernel used to estimate the conditional distribution function.

Usage

```
kernel.function.quart(u)
```

Arguments

u   a numeric vector.

Author(s)

Romain Azais, Sandie Ferrigno and Marie-Jose Martinez

References

J. Fan and I. Gijbels. Local polynomial modelling and its applications. *Chapman & Hall*, London, 1996.

Examples

```
x = runif(10)  # simulating a uniform random sample
kernel.function.quart(x)
```

---

vkgmss.bandwidth.selection.linkfunction
Bandwidth selection of the link function under the null hypothesis

---

Description

This function computes the optimal bandwidth of the link function under the null hypothesis

Usage

```
vkgmss.bandwidth.selection.linkfunction(data.X.H0, data.Y.H0, linkfunction.H0,
kernel.function = kernel.function.epan, verbose = TRUE)
```

Arguments

data.X.H0	a numeric data vector used to obtain the nonparametric estimator of the regression function under the null hypothesis.
data.Y.H0	a numeric data vector used to obtain the nonparametric estimator of the regression function under the null hypothesis.
linkfunction.H0	
	regression function under the null hypothesis
kernel.function	
	kernel function used to obtain the nonparametric estimator of the regression function. Default option is "kernel.function.epan" which corresponds to the Epanechnikov kernel function.
verbose	If TRUE, the R function plots the link function (regression function) under the null hypothesis and the nonparametric link function estimation on a same graph. Default option is TRUE.

Author(s)

Romain Azais, Sandie Ferrigno and Marie-Jose Martinez

References

I. Van Keilegom, W. Gonzalez Manteiga, and C. Sanchez Sellero. Goodness-of-fit tests in parametric regression based on the estimation of the error distribution. *Test*, 17, 401:415, 2008.

R. Azais, S. Ferrigno and M-J Martinez. cvmgof: An R package for Cramér-von Mises goodness-of-fit tests in regression models. 2018. Preprint in progress.

Examples

```
set.seed(1)

# Data simulation
n = 25 # Dataset size
data.X = runif(n,min=0,max=5) # X
data.Y = 0.2*data.X^2-data.X+2+rnorm(n,mean=0,sd=0.3) # Y

#########################################################################

# Bandwidth selection under H0

# We generate a dataset under H0 to estimate the optimal bandwidth under H0

linkfunction.H0 = function(x){0.2*x^2-x+2}

data.X.H0 = runif(n,min=0,max=5)
data.Y.H0 = linkfunction.H0(data.X.H0)+rnorm(n,mean=0,sd=0.3)

h.opt.vkgmss = vkgmss.bandwidth.selection.linkfunction(data.X.H0 , data.Y.H0,linkfunction.H0)
```

---

vkgmss.linkfunction.estim

Kernel estimation of the regression function

Description

This function computes the kernel (Nadaraya-Watson) estimation of the regression function.

Usage

```
vkgmss.linkfunction.estim(x, data.X, data.Y, bandwidth,
kernel.function = kernel.function.epan)
```

Arguments

x	a numeric vector.
data.X	a numeric data vector used to obtain the kernel estimator of the regression function.

data.Y	a numeric data vector used to obtain the kernel estimator of the regression function.
bandwidth	bandwidth used to obtain the kernel estimator of the regression function.
kernel.function	
	kernel function used to obtain the kernel estimator of the regression function. Default option is "kernel.function.epan" which corresponds to the Epanechnikov kernel function.

Details

Inappropriate bandwidth or x choices can produce "NaN" values in link function estimates.

Author(s)

Romain Azais, Sandie Ferrigno and Marie-Jose Martinez

References

I. Van Keilegom, W. Gonzalez Manteiga, and C. Sanchez Sellero. Goodness-of-fit tests in parametric regression based on the estimation of the error distribution. *Test*, 17, 401:415, 2008.

R. Azais, S. Ferrigno and M-J Martinez. cvmgof: An R package for Cramér-von Mises goodness-of-fit tests in regression models. 2018. Preprint in progress.

Examples

```
set.seed(1)

# Data simulation
n = 25 # Dataset size
data.X = runif(n,min=0,max=5) # X
data.Y = 0.2*data.X^2-data.X+2+rnorm(n,mean=0,sd=0.3) # Y

#########################################################################

# Estimation of the link function

bandwidth = 0.75 # Here, the bandwidth is arbitrarily fixed

xgrid = seq(0,5,by=0.1)
ygrid_vkgmss = vkgmss.linkfunction.estim(xgrid,data.X,data.Y,bandwidth)

plot(xgrid,ygrid_vkgmss,type='l',lty=1,lwd=2,xlab='X',ylab='Y',ylim=c(0.25,2.5))
lines(xgrid,0.2*xgrid^2-xgrid+2,lwd=0.5,col='gray')
```

---

vkgmss.residuals.cdf.estim

Kernel estimation of the error distribution

Description

This function computes the kernel (Nadaraya-Watson) estimation of the error distribution.

Usage

```
vkgmss.residuals.cdf.estim(u, data.X, data.Y, bandwidth,
kernel.function = kernel.function.epan)
```

Arguments

u	a numeric vector.
data.X	a numeric data vector used to obtain the nonparametric estimator of the error distribution.
data.Y	a numeric data vector used to obtain the nonparametric estimator of the error distribution.
bandwidth	bandwidth used to obtain the nonparametric estimator of the error distribution.
kernel.function	
	kernel function used to obtain the nonparametric estimator of the error distribution. Default option is "kernel.function.epan" which corresponds to the Epanechnikov kernel function.

Details

Inappropriate bandwidth or u choices can produce "NaN" values in error distribution estimates.

Author(s)

Romain Azais, Sandie Ferrigno and Marie-Jose Martinez

References

I. Van Keilegom, W. Gonzalez Manteiga, and C. Sanchez Sellero. Goodness-of-fit tests in parametric regression based on the estimation of the error distribution. *Test*, 17, 401:415, 2008.

R. Azais, S. Ferrigno and M-J Martinez. cvmgof: An R package for Cramér-von Mises goodness-of-fit tests in regression models. 2018. Preprint in progress.

Examples

```
set.seed(1)

# Data simulation
n = 25 # Dataset size
data.X = runif(n,min=0,max=5) # X
data.Y = 0.2*data.X^2-data.X+2+rnorm(n,mean=0,sd=0.3) # Y

########################################################################

# Estimation of residuals cdf

bandwidth = 0.75 # Here, the bandwidth is arbitrarily fixed

egrid = seq(-5,5,by=0.1)
res.cdf = vkgmss.residuals.cdf.estim(egrid,data.X,data.Y,0.5)

plot(egrid,res.cdf , type='l',xlab='e',ylab='CDF(e)')
```

---

vkgmss.sd.estim   *Kernel estimation of the standard deviation function*

Description

This function computes the kernel (Nadaraya-Watson) estimation of the standard deviation function.

Usage

```
vkgmss.sd.estim(x, data.X, data.Y, bandwidth,
kernel.function = kernel.function.epan)
```

Arguments

x	a numeric vector.
data.X	a numeric data vector used to obtain the kernel estimator of the standard deviation function.
data.Y	a numeric data vector used to obtain the kernel estimator of the standard deviation function.
bandwidth	bandwidth used to obtain the kernel estimator of the standard deviation function.
kernel.function	kernel function used to obtain the kernel estimator of the standard deviation function. Default option is "kernel.function.epan" which corresponds to the Epanechnikov kernel function.

Details

Inappropriate bandwidth or x choices can produce "NaN" values in function estimates.

Author(s)

Romain Azais, Sandie Ferrigno and Marie-Jose Martinez

References

I. Van Keilegom, W. Gonzalez Manteiga, and C. Sanchez Sellero. Goodness-of-fit tests in parametric regression based on the estimation of the error distribution. *Test*, 17, 401:415, 2008.

R. Azais, S. Ferrigno and M-J Martinez. cvmgof: An R package for Cramér-von Mises goodness-of-fit tests in regression models. 2018. Preprint in progress.

Examples

```
set.seed(1)

# Data simulation
n = 25 # Dataset size
data.X = runif(n,min=0,max=5) # X
data.Y = 0.2*data.X^2-data.X+2+rnorm(n,mean=0,sd=0.3) # Y

#########################################################################

# Estimation of residuals standard deviation

bandwidth = 0.75 # Here, the bandwidth is arbitrarily fixed

xgrid = seq(0,5,by=0.1)
sd = vkgmss.sd.estim(xgrid,data.X,data.Y,bandwidth)

plot(xgrid,sd , type='l',xlab='X',ylab='SD(X)')
abline(h=0.3)
```

---

vkgmss.statistics   *Local test statistic for the regression function*

Description

This function computes the local test statistic for the regression function.

Usage

```
vkgmss.statistics(data.X, data.Y, linkfunction.H0,
bandwidth = "optimal", kernel.function = kernel.function.epan,
verbose=TRUE)
```

Arguments

`data.X`	a numeric data vector used to obtain the nonparametric estimator of the error distribution.
`data.Y`	a numeric data vector used to obtain the nonparametric estimator of the error distribution.
`linkfunction.H0`	
	the regression function under the null hypothesis.
`bandwidth`	bandwidth used to obtain the nonparametric estimator of the error distribution. If `bandwidth="optimal"`, the optimal bandwidth of the regression function under the null hypothesis is computed. Default option is "optimal".
`kernel.function`	
	kernel function used to obtain the nonparametric estimator of the error distribution. Default option is "kernel.function.epan".
`verbose`	If TRUE, the R function displays the optimal bandwidth value obtained under the null hypothesis. Default option is TRUE.

Author(s)

Romain Azais, Sandie Ferrigno and Marie-Jose Martinez

References

I. Van Keilegom, W. Gonzalez Manteiga, and C. Sanchez Sellero. Goodness-of-fit tests in parametric regression based on the estimation of the error distribution. *Test*, 17, 401:415, 2008.

R. Azais, S. Ferrigno and M-J Martinez. cvmgof: An R package for Cramér-von Mises goodness-of-fit tests in regression models. 2018. Preprint in progress.

Examples

```
set.seed(1)

# Data simulation
n = 25 # Dataset size
data.X = runif(n,min=0,max=5) # X
data.Y = 0.2*data.X^2-data.X+2+rnorm(n,mean=0,sd=0.3) # Y

##########################################################################

# Bandwidth selection under H0

# We want to test if the link function is f(x)=0.2*x^2-x+2
# The answer is yes (see the definition of data.Y above)
# We generate a dataset under H0 to estimate the optimal bandwidth under H0

linkfunction.H0 = function(x){0.2*x^2-x+2}

data.X.H0 = runif(n,min=0,max=5)
data.Y.H0 = linkfunction.H0(data.X.H0)+rnorm(n,mean=0,sd=0.3)
```

```
h.opt.vkgmss = vkgmss.bandwidth.selection.linkfunction(data.X.H0 , data.Y.H0,linkfunction.H0)

#######################################################################

# Test statistics under H0

vkgmss.statistics(data.X,data.Y,linkfunction.H0,h.opt.vkgmss)
```

---

vkgmss.test.bootstrap   *Local test for the regression function*

Description

A local test for the regression function.

Usage

```
vkgmss.test.bootstrap(data.X, data.Y, linkfunction.H0, risk,
    bandwidth = "optimal", kernel.function = kernel.function.epan,
    bootstrap = c(50, "Mammen"), verbose = TRUE)
```

Arguments

data.X	a numeric data vector used to obtain the nonparametric estimator of the error distribution.
data.Y	a numeric data vector used to obtain the nonparametric estimator of the error distribution.
linkfunction.H0	
	the regression function under the null hypothesis.
risk	a numeric value specifying the risk of rejecting the null hypothesis. The value (1-risk) corresponds to the confidence level of the statistical test.
bandwidth	the bandwidth used to obtain the nonparametric estimator of the error distribution. If bandwidth="optimal", the optimal bandwidth of the regression function under the null hypothesis is computed. Default option is "optimal".
kernel.function	
	the kernel function used to obtain the nonparametric estimator of the error distribution. Default option is "kernel.function.epan".
bootstrap	a numeric vector of length 2. The first value specifies the number of bootstrap datasets (default is "50"). The second value specifies the distribution used for the wild bootstrap resampling (default is "Mammen").
verbose	If TRUE, the R function displays the optimal bandwidth value obtained under the null hypothesis. Default option is TRUE.

Details

From `data.X` and `data.Y` datasets, wild bootstrap datasets ("50" by default) are built. From each bootstrap dataset, a bootstrap test statistic is computed. The test statistic under the null hypothesis is compared to the distribution of the bootstrap statistics. The test is rejected if the test statistic under the null hypothesis is greater than the (`1-risk`)-quantile of the empirical distribution of the bootstrap statistics.

An inappropriate bandwidth choice can produce "NaN" values in test statistics.

Value

`vkgmss.test.bootstrap` returns a list containing the following components:

`decision`	the statistical decision made on whether to reject the null hypothesis or not.
`bandwidth`	the bandwidth used to build the statistics test.
`pvalue`	the p-value of the test statistics.
`test_statistics`	
	the test statistics value.

Author(s)

Romain Azais, Sandie Ferrigno and Marie-Jose Martinez

References

I. Van Keilegom, W. Gonzalez Manteiga, and C. Sanchez Sellero. Goodness-of-fit tests in parametric regression based on the estimation of the error distribution. *Test*, 17, 401:415, 2008.

R. Azais, S. Ferrigno and M-J Martinez. cvmgof: An R package for Cramér-von Mises goodness-of-fit tests in regression models. 2018. Preprint in progress.

Examples

```
set.seed(1)

# Data simulation
n = 25 # Dataset size
data.X = runif(n,min=0,max=5) # X
data.Y = 0.2*data.X^2-data.X+2+rnorm(n,mean=0,sd=0.3) # Y

#########################################################################

# Test (bootstrap) under H0

# We want to test if the link function is f(x)=0.2*x^2-x+2
# The answer is yes (see the definition of data.Y above)
# We generate a dataset under H0 to estimate the optimal bandwidth under H0

linkfunction.H0 = function(x){0.2*x^2-x+2}

test_vkgmss.H0 = vkgmss.test.bootstrap(data.X,data.Y,linkfunction.H0,
```

```
0.05,bandwidth='optimal',bootstrap=c(50,'Mammen'))


#######################################################################

# Test (bootstrap) under H1

# We want to test if the link function is f(x)=0.5*cos(x)+1
# The answer is no (see the definition of data.Y above)

linkfunction.H1=function(x){0.8*cos(x)+1}

test_vkgmss.H1 = vkgmss.test.bootstrap(data.X,data.Y,linkfunction.H1,
 0.05,bandwidth='optimal',bootstrap=c(50,'Mammen'))
```

Index