Package ‘sROC’

February 20, 2015

**Type**  Package

**Title**  Nonparametric Smooth ROC Curves for Continuous Data

**Version**  0.1-2

**Date**  2012-04-12

**Author**  Xiao-Feng Wang

**Maintainer**  Xiao-Feng Wang <wangx@ccf.org>

**Description**  This package contains a collection of functions to perform nonparametric estimation of receiver operating characteristic (ROC) curves for continuous data.

**License**  GPL (>= 3)

**Repository**  CRAN

**Date/Publication**  2012-04-12 14:26:33

**NeedsCompilation**  yes

### R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUC</td>
<td>2</td>
</tr>
<tr>
<td>bw.CDF</td>
<td>3</td>
</tr>
<tr>
<td>bw.CDF.pi</td>
<td>4</td>
</tr>
<tr>
<td>CL.CDF</td>
<td>5</td>
</tr>
<tr>
<td>kCDF</td>
<td>6</td>
</tr>
<tr>
<td>kROC</td>
<td>8</td>
</tr>
<tr>
<td>plot.CDF</td>
<td>10</td>
</tr>
<tr>
<td>plot.ROC</td>
<td>11</td>
</tr>
<tr>
<td>print.AUC</td>
<td>12</td>
</tr>
<tr>
<td>print.CDF</td>
<td>12</td>
</tr>
<tr>
<td>print.ROC</td>
<td>13</td>
</tr>
</tbody>
</table>

**Index**  14
**AUC**

**Area Under Curve**

**Description**

Compute the area under curve of estimated ROC curve.

**Usage**

```r
AUC(ROC, method="Simpson", ngrid=256)
```

**Arguments**

- `ROC` a “ROC” object generated by `kROC(...)``
- `method` a character string giving the numerical integration method to be used. This must be either “Simpson” or “Trapez”.
- `ngrid` the number of grids for numerical integration.

**Details**

Compute the area under curve of estimated ROC curve.

**Value**

An object of class “AUC”.

**Author(s)**

X.F. Wang <wangx6@ccf.org>

**See Also**

`kROC`.

**Examples**

```r
set.seed(100)
n <- 200
x <- rlnorm(n, mean=2, sd=1)
y <- rnorm(n, mean=2, sd=2)

xy.ROC <- kROC(c(x,NA,NA),c(y,1.2,NA), na.rm=TRUE)
plot(xy.ROC)
AUC(xy.ROC)
```
bw.CDF

Bandwidth Selectors for Kernel CDF Estimation

Description

Rule-of-thumb bandwidth selectors for kernel CDF estimation using the normal CDF or PDF reference approach.

Usage

bw.CDF(x, method="npdf")

Arguments

x numeric vector.
method either “npdf” (the normal PDF reference approach) or “ncdf” (the normal CDF reference approach).

Details

bw.CDF implements a rule-of-thumb for choosing the bandwidth of a Gaussian kernel CDF estimator.

Value

A bandwidth on a scale suitable for the bw argument of kCDF.

Author(s)

X.F. Wang <wangx6@ccf.org>

References


See Also

kCDF, bw.CDF.pi.

Examples

set.seed(100)
n <- 200
x <- c(rnorm(n/2, mean=-2, sd=1), rnorm(n/2, mean=3, sd=0.8))
bw.CDF(x, method="npdf")
bw.CDF(x, method="ncdf")
Plug-in Bandwidth Selectors for Kernel CDF Estimation

Description
Plug-in bandwidth selectors for kernel CDF estimation using Altman and Leger’s approach.

Usage
bw.CDF.pi(x, pilot="ucv")

Arguments
x numeric vector.
pilot a character string giving a rule to choose the pilot bandwidth to estimate \( E(f''(x)) \).

There are 6 choices: The default, “ucv”, implement unbiased cross-validation, “nrd0” is the Silverman’s rule of thumb, “nrd” is the Scott’s method (1992), “bcv” is the biased cross-validation, “sj” is the method of Sheather and Jones (1991), “onestage” is the one-stage method by Wand and Jones.

Details
bw.CDF implements a rule-of-thumb for choosing the bandwidth of a Gaussian kernel CDF estimator.

Value
A bandwidth on a scale suitable for the bw argument of kCDF.

Author(s)
X.F. Wang <wangx6@ccf.org>

References

See Also
kCDF, bw.CDF.
Examples

```r
set.seed(100)
n <- 200
x <- c(rnorm(n/2, mean=-2, sd=1), rnorm(n/2, mean=3, sd=0.8))
bw.CDF.pi(x)
bw.CDF.pi(x, pilot="nrd0")
bw.CDF.pi(x, pilot="nrd")
bw.CDF.pi(x, pilot="bcv")
bw.CDF.pi(x, pilot="sj")
bw.CDF.pi(x, pilot="onestage")
```

---

**CI.CDF**

*Pointwise Confidence Intervals for Kernel Smooth CDF*

**Description**

Estimate the pointwise confidence intervals for Kernel Smooth CDF.

**Usage**

```r
CI.CDF(CDF, alpha=0.05)
```

**Arguments**

- **CDF**: a “CDF” object generated by kCDF(...).
- **alpha**: the significant level. The default is 0.05 which generates 95% confidence intervals for the CDF.

**Details**

The pointwise confidence intervals are calculated by the asymptotic distribution of the kernel estimator of CDF.

**Value**

A list contents

- **x**: the points where the CDF is estimated.
- **Fhat**: the estimated CDF values. These will be numerical numbers between zero and one.
- **Fhat.upper**: the upper boundaries of the CDF.
- **Fhat.lower**: the lower boundaries of the CDF.
- **alpha**: the significant level used.
Author(s)
X.F. Wang <wangx6@ccf.org>

References


See Also
kcdf, bw.CDF.pi.

Examples

```r
set.seed(100)
n <- 200
x <- c(rnorm(n/2, mean=-2, sd=1), rnorm(n/2, mean=3, sd=0.8))
x.CDF <- kcdf(x)
x.CDF
CI.CDF(x.CDF)
plot(x.CDF, alpha=0.05, main="Kernel estimate of distribution function")
curve(pnorm(x, mean=-2, sd=1)/2 + pnorm(x, mean=3, sd=0.8)/2, from = -6, to = 6, add = TRUE, lty = 2, col = "blue")
```

---

**kcdf**

*Kernel Estimation for Cumulative Distribution Function*

**Description**

To compute the nonparametric kernel estimate for cumulative distribution function (CDF).

**Usage**

```r
kcdf(x, bw="pi_ucv", adjust=1, kernel=c("normal", "epanechnikov"), xgrid, ngrid=256, from, to, cut=3, na.rm = FALSE, ...)
```

**Arguments**

- `x`  the data from which the estimate is to be computed.
- `bw`  the smoothing bandwidth to be used. `bw` can also be a character string giving a rule to choose the bandwidth. See `bw.CDF` and `bw.CDF.pi`. The default used the Altman and Leger’s plug-in approach with an unbiased cross-validation pilot bandwidth.
adjust  the parameter for adjusting the bandwidth. The bandwidth used for the estimate is actually adjust*bw. By default, adjust = 1.

kernel  a character string giving the smoothing kernel to be used. This must be either “normal” or “epanechnikov”. By default, the normal kernel is used.

xgrid  the user-defined data points at which the CDF is to be evaluated. If missing, the CDF will be evaluated at the equally spaced points defined within the function.

ngrid  the number of equally spaced points at which the density is to be estimated.

from  the left-most points of the grid at which the density is to be estimated.

to  the right-most points of the grid at which the density is to be estimated.

cut  by default, the values of from and to are cut bandwidths beyond the extremes of the data.

na.rm  logical; if TRUE, missing values are removed from x. If FALSE any missing values cause an error.

...  further arguments for methods.

Details

estimate the nonparametric kernel cumulative distribution function.

Value

An object of class “CDF”.

x  the points where the CDF is estimated.

fhat  the estimated CDF values. These will be numerical numbers between zero and one.

bw  the bandwidth used.

n  the sample size after elimination of missing values.

call  the call which produced the result.

data  the original data after elimination of missing values.

data.name  the deparsed name of the x argument.

has.na  logical; if TRUE, there are missing values in the original data.

The print method reports summary values on the x and fhat components.

Author(s)

X.F. Wang <wangx6@ccf.org>

References


See Also

bw.CDF, bw.CDF.pi.

Examples

```r
## -------------------
set.seed(100)
N <- 200
x <- c(rnorm(N/2, mean=-2, sd=1), rnorm(N/2, mean=3, sd=0.8))
x.CDF <- kCDF(x)
x.CDF
plot(x.CDF, alpha=0.05, main="Kernel estimate of distribution function", CI=FALSE)
curve(pnorm(x, mean=-2, sd=1)/2 + pnorm(x, mean=3, sd=0.8)/2, from =-6, to=6, add=TRUE, lty=2, col="blue")
```

---

kROC

**Kernel Estimation for ROC Curves**

**Description**

To compute the nonparametric kernel estimate of receiver operating characteristic (ROC) Curves for continuous data.

**Usage**

```r
kROC(x, y, bw.x="pi_ucv", bw.y="pi_ucv", adjust=1, kernel=c("normal", "epanechnikov"), xgrid, ngrid=256, from, to, cut=3, na.rm = FALSE, ...)
```

**Arguments**

- `x` numeric vector.
- `y` numeric vector.
- `bw.x` the smoothing bandwidth of `x` to be used. `bw` can also be a character string giving a rule to choose the bandwidth. See `bw.CDF` and `bw.CDF.pi`. The default used the Altman and Leger’s plug-in approach with an unbiased cross-validation pilot bandwidth.
- `bw.y` the smoothing bandwidth of `y` to be used.
- `adjust` the parameter for adjusting the bandwidth. The bandwidth used for the estimate is actually `adjust*bw`. By default, `adjust = 1`.
- `kernel` a character string giving the smoothing kernel to be used. This must be either “normal” or “epanechnikov”. By default, the normal kernel is used.
- `xgrid` the user-defined data points at which the CDF is to be evaluated. If missing, the CDF will be evaluated at the equally spaced points defined within the function.
- `ngrid` the number of equally spaced points at which the density is to be estimated.
from the left-most points of the grid at which the density is to be estimated.
to the right-most points of the grid at which the density is to be estimated
cut by default, the values of from and to are cut bandwidths beyond the extremes of
the data.
na.rm logical; if TRUE, missing values are removed from x. If FALSE any missing values
cause an error.
... further arguments for methods.

Details

estimate the nonparametric kernel estimate of receiver operating characteristic (ROC) Curves for
continuous data

Value

An object of class “ROC”.

FPR the false positive rate.
TPR the true positive rate.
bw.x, bw.y the bandwidths used.
nx, ny the sample sizes after elimination of missing values.
call the call which produced the result.
x.data.name, y.data.name
the deparsed names of the x argument.
x.has.na, y.has.na
logical; if TRUE, there are missing values in the original data.

The print method reports summary values on the x and Fhat components.

Author(s)

X.F. Wang <wangx6@ccf.org>

References

Lloyd, C.J. (1998). Using smoothed receiver operating characteristic curves to summarize and

Zhou, X.H. and Harezlak, J. (2002). Comparison of bandwidth selection methods for kernel smooth-

Zou, K.H., Hall, W.J., and Shapiro, D.E. (1997). Smooth non-parametric receiver operating char-

See Also

bw.CDF, bw.CDF.pi.
Examples

```r
## ------------------------
set.seed(100)
n <- 200
x <- rgamma(n,2,1)
y <- rnorm(n)

xy.ROC <- kROC(x,y, bw.x="pi_sj",bw.y="pi_sj")
xy.ROC

plot(xy.ROC)
```

---

**plot.CDF**

*Plot a CDF Object*

Description

To plot a “CDF” object generated by kCDF(...).

Usage

```r
## S3 method for class 'CDF'
plot(x, CI=TRUE, alpha=0.05, main = NULL, xlab = NULL, ylab = "cdf", lwd=2, lty=1, ...)
```

Arguments

- **x**: a “CDF” object generated by kCDF(...).
- **CI**: If TRUE, the pointwise confidence intervals will be plotted.
- **alpha**: the significant level. The default is 0.05 which generates 95% confidence intervals for the CDF.
- **main**: see `par`
- **xlab**: see `par`
- **ylab**: see `par`
- **lwd**: see `par`
- **lty**: see `par`
- **...**: further arguments for the plot function.

Details

This function is to plot the estimated function generated by kCDF(...)

Author(s)

X.F. Wang <wangx6@ccf.org>
Description

To plot a “ROC” object generated by kROC(...).

Usage

```r
## S3 method for class 'ROC'
plot(x, main = NULL, diagonal = TRUE, xlab = "FPR", ylab = "TPR", type = "l", lwd = 2, ...)
```

Arguments

- `x`: a “ROC” object generated by kROC(...).
- `diagonal`: if TRUE, the diagonal line will be plotted.
- `main`: see `par`
- `xlab`: see `par`
- `ylab`: see `par`
- `type`: see `par`
- `lwd`: see `par`
- `...`: further arguments for the plot function.

Details

This function is to plot the estimated function generated by kROC(...)

Author(s)

X.F. Wang <wangx6@ccf.org>

See Also

- kROC.
print.AUC  Print a AUC Object

Description
To print a “AUC” object generated by AUC(…).

Usage
## S3 method for class 'AUC'
print(x, digits = NULL, ...)

Arguments
- x: a “AUC” object generated by AUC(…).
- digits: integer indicating the number of decimal places to be used.
- ...: further arguments for the print function.

Details
This function is to print the summary description from the object generated by AUC(…)

Author(s)
X.F. Wang <wangx6@ccf.org>

See Also
- AUC.

print.CDF  Print a CDF Object

Description
To print a “CDF” object generated by kCDF(…).

Usage
## S3 method for class 'CDF'
print(x, digits = NULL, ...)

Arguments
- x: a “CDF” object generated by kCDF(…).
- digits: integer indicating the number of decimal places to be used.
- ...: further arguments for the print function.
print.ROC

Arguments

x a "CDF" object generated by kCDF(...).
digits integer indicating the number of decimal places to be used.
... further arguments for the print function.

Details

This function is to print the summary description from the object generated by kCDF(...).

Author(s)

X.F. Wang <wangx6@ccf.org>

See Also

kCDF.

print.ROC  Print a ROC Object

Description

To print a "ROC" object generated by kROC(...).

Usage

## S3 method for class 'ROC'
print(x, digits = NULL, ...)

Arguments

x a "ROC" object generated by kROC(...).
digits integer indicating the number of decimal places to be used.
... further arguments for the print function.

Details

This function is to print the summary description from the object generated by kROC(...).

Author(s)

X.F. Wang <wangx6@ccf.org>

See Also

kROC.
Index

*Topic distribution function estimation
  kCDF. 6
  nonparametric
  kCDF. 6
  kROC. 8
*Topic smoothing
  AUC. 2
  bw.CDF. 3
  bw.CDF.pi. 4
  CI.CDF. 5
  plot.CDF. 10
  plot.ROC. 11
  print.AUC. 12
  print.CDF. 12
  print.ROC. 13
*Topic smooth
  kCDF. 6
  kROC. 8

AUC. 2, 12

bw.CDF. 3, 4, 6, 8, 9
bw.CDF.pi. 3, 4, 6, 8, 9

CI.CDF. 5

kCDF. 3, 4, 6, 6, 11, 13
kROC. 2, 8, 11, 13

par. 10, 11

plot.CDF. 10
plot.ROC. 11
print.AUC. 12
print.CDF. 12
print.ROC. 13

summary. 7, 9