Package: rbbnp (via r-universe)

June 9, 2026

Type Package

Title A Bias Bound Approach to Non-Parametric Inference

Version 1.1.0

Description A novel bias-bound approach for non-parametric inference is introduced, focusing on both density and conditional expectation estimation. It constructs valid confidence intervals that account for the presence of a non-negligible bias and thus make it possible to perform inference with optimal mean squared error minimizing bandwidths. This package is based on Schennach (2020) <doi:10.1093/restud/rdz065>.

License GPL (>= 3)

Encoding UTF-8

LazyData true

URL https://xinyudai.net/rbbnp-dev/

Imports dplyr, ggplot2 (>= 3.4.0), gridExtra, pracma, purrr, Rglpk, tidyr

RoxygenNote 7.3.3

Depends R (>= 3.5)

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

NeedsCompilation no

Author Xinyu DAI [aut, cre], Susanne M Schennach [aut]

Maintainer Xinyu DAI <xinyu_dai@brown.edu>

Repository https://cran.r-universe.dev

Date/Publication 2026-06-09 04:40:36 UTC

RemoteUrl https://github.com/cran/rbbnp

RemoteRef HEAD

RemoteSha 9e4958802345bb7421eeca31a09e2183c27acb5f

Contents

`biasBound_condExpectation`

Bias bound approach for conditional expectation estimation

Description

Estimates the density at a given point or across a range, and provides visualization options for density, bias, and confidence intervals.

Usage

```
biasBound_condExpectation(
  Y,
  X,
  x = NULL,
  h = NULL,
  h_method = "cv",
  alpha = 0.05,
  est_Ar = NULL,
  resol = 100,
  xi_lb = NULL,
  xi_ub = NULL,
  methods_get_xi = "snr",
```

```
  noise_floor = "auto",
  envelope_use_Y = TRUE,
  integer_r = TRUE,
  ora_Ar = NULL,
  kernel.fun = "Schennach2004",
  if_approx_kernel = TRUE,
  kernel.resol = 1000
)
```

Arguments

Y	A numerical vector of sample data.
X	A numerical vector of sample data.
x	Optional. A scalar or range of points where the density is estimated. If NULL, a range is automatically generated.
h	A scalar bandwidth parameter. If NULL, the bandwidth is automatically selected using the method specified in 'h_method'.
h_method	Method for automatic bandwidth selection when h is NULL. Options are "cv" (cross-validation) and "silverman" (Silverman's rule of thumb). Default is "cv".
alpha	Confidence level for intervals. Default is 0.05.
est_Ar	Optional list of estimates for A and r. If NULL, they are computed using `get_est_Ar()`.
resol	Resolution for the estimation range. Default is 100.
xi_lb	Optional. Lower bound for the interval of Fourier Transform frequency xi. Used for determining the range over which A and r is estimated. If NULL, it is automatically determined based on the methods_get_xi.
xi_ub	Optional. Upper bound for the interval of Fourier Transform frequency xi. Similar to xi_lb, it defines the upper range for A and r estimation. If NULL, the upper bound is determined based on the methods_get_xi.
methods_get_xi	A string selecting the frequency-window rule used when xi_lb/xi_ub are NULL: "snr" (default; a signal-to-noise cutoff that selects a valid window at realistic sample sizes), "Schennach" (the data-driven rule of Schennach 2020, Theorem 2), or "Schennach_loose" (the initial, un-refined interval).
noise_floor	Noise-floor form for the Schennach test: "auto" (default), "compact", or "general".
envelope_use_Y	If TRUE (default), fit the regression envelope to the cross-spectrum `|phi_YX|`; if FALSE, fit it to the marginal spectrum `|phi_X|`.
integer_r	If TRUE (default), clamp the fitted envelope slope up to r = 2 when it falls below the minimum smoothness assumed by Schennach (2020, Definition 2), i.e. r < 2, and refit A; this keeps the bias-bound integral finite. Slopes >= 2 are left unchanged.
ora_Ar	Optional list of oracle values for A and r (for research/comparison purposes).
kernel.fun	A string specifying the kernel function to be used. Options are "Schennach2004", "sinc", "normal", "epanechnikov".

`if_approx_kernel`
: Logical. If TRUE, uses approximations for the kernel function.

`kernel.resol`
: The resolution for kernel function approximation. See `fun_approx`.

Value

An object of class `bbnp_regression` with components:

`fitted_values`
: $E[Y|X = x]$ estimates (for range estimation)

`x`
: Evaluation points

`estimate`
: Point estimate (for single x)

`conf_int`
: List containing lower, upper bounds and conf_level. Note that the confidence interval can be unbounded (i.e., contain `-Inf` or `Inf`) in regions where the estimated marginal density $\hat{f}(x)$ is very close to zero, because the estimator is formed as a ratio involving $1/\hat{f}(x)$.

`bias_bound`
: List containing b1x, byx, est_A, est_r, est_B, xi_interval

`std_error`
: Standard errors

`marginal_density`
: f(x) estimates

`joint_density`
: f_YX estimates

`call`
: The function call

`bandwidth`
: Bandwidth used

`n`
: Sample size

`kernel`
: Kernel type

`data`
: Original data (X, Y)

Use `plot()`, `summary()`, `coef()`, `fitted()`, and `confint()` methods to work with the result.

Examples

```
# Example 1: Point estimation at x = 1
X <- rnorm(100)
Y <- X^2 + rnorm(100)
fit <- biasBound_condExpectation(Y = Y, X = X, x = 1, h = 0.09)
print(fit)
fitted(fit)

# Example 2: Range estimation with plots
fit2 <- biasBound_condExpectation(Y = Y, X = X, h = NULL, h_method = "cv")
plot(fit2)              # Regression plot
plot(fit2, type = "ft") # Fourier transform plot
summary(fit2)
```

biasBound_density *Bias bound approach for density estimation*

Description

Estimates the density at a given point or across a range, and provides visualization options for density, bias, and confidence intervals.

Usage

```
biasBound_density(
  X,
  x = NULL,
  h = NULL,
  h_method = "cv",
  alpha = 0.05,
  resol = 100,
  xi_lb = NULL,
  xi_ub = NULL,
  methods_get_xi = "snr",
  noise_floor = "auto",
  envelope_use_Y = TRUE,
  integer_r = TRUE,
  ora_Ar = NULL,
  kernel.fun = "Schennach2004",
  if_approx_kernel = TRUE,
  kernel.resol = 1000
)
```

Arguments

X	A numerical vector of sample data.
x	Optional. A scalar or range of points where the density is estimated. If NULL, a range is automatically generated.
h	A scalar bandwidth parameter. If NULL, the bandwidth is automatically selected using the method specified in 'h_method'.
h_method	Method for automatic bandwidth selection when h is NULL. Options are "cv" (cross-validation) and "silverman" (Silverman's rule of thumb). Default is "cv".
alpha	Confidence level for intervals. Default is 0.05.
resol	Resolution for the estimation range. Default is 100.
xi_lb	Optional. Lower bound for the interval of Fourier Transform frequency xi. Used for determining the range over which A and r is estimated. If NULL, it is automatically determined based on the methods_get_xi.
xi_ub	Optional. Upper bound for the interval of Fourier Transform frequency xi. Similar to xi_lb, it defines the upper range for A and r estimation. If NULL, the upper bound is determined based on the methods_get_xi.

methods_get_xi	A string selecting the frequency-window rule used when xi_lb/xi_ub are NULL: "snr" (default; a signal-to-noise cutoff that selects a valid window at realistic sample sizes), "Schennach" (the data-driven rule of Schennach 2020, Theorem 2), or "Schennach_loose" (the initial, un-refined interval).
noise_floor	Noise-floor form for the Schennach test: "auto" (default), "compact", or "general".
envelope_use_Y	If TRUE (default), fit the regression envelope to the cross-spectrum |phi_YX|; if FALSE, fit it to the marginal spectrum |phi_X|.
integer_r	If TRUE (default), clamp the fitted envelope slope up to r = 2 when it falls below the minimum smoothness assumed by Schennach (2020, Definition 2), i.e. r < 2, and refit A; this keeps the bias-bound integral finite. Slopes >= 2 are left unchanged.
ora_Ar	Optional list of oracle values for A and r (for research/comparison purposes).
kernel.fun	A string specifying the kernel function to be used. Options are "Schennach2004", "sinc", "normal", "epanechnikov".
if_approx_kernel	Logical. If TRUE, uses approximations for the kernel function.
kernel.resol	The resolution for kernel function approximation. See fun_approx.

Value

An object of class bbnp_density with components:

density	Density estimates (for range estimation)
x	Evaluation points
estimate	Point estimate (for single x)
conf_int	List containing lower, upper bounds and conf_level
bias_bound	List containing b1x, est_A, est_r, xi_interval
std_error	Standard errors
call	The function call
bandwidth	Bandwidth used
n	Sample size
kernel	Kernel type
data	Original data

Use plot(), summary(), coef(), and confint() methods to work with the result.

Examples

```
# Example 1: Point estimation at x = 1
X <- rnorm(100)
fit <- biasBound_density(X = X, x = 1, h = 0.09)
print(fit)
coef(fit)
```

```
# Example 2: Range estimation with automatic bandwidth
fit2 <- biasBound_density(X = X, h = NULL, h_method = "cv")
plot(fit2)           # Density plot
plot(fit2, type = "ft")  # Fourier transform plot
summary(fit2)
```

---

coef.bbnp_density *Extract Coefficients from bbnp_density Object*

Description

Extracts the estimated bias bound parameters and bandwidth

Usage

```
## S3 method for class 'bbnp_density'
coef(object, ...)
```

Arguments

object	An object of class bbnp_density
...	Additional arguments (unused)

Value

Named numeric vector with A (amplitude), r (decay rate), and h (bandwidth)

Examples

```
X <- rnorm(100)
fit <- biasBound_density(X, h = 0.1)
coef(fit)
```

---

coef.bbnp_regression *Extract Coefficients from bbnp_regression Object*

Description

Extracts the estimated bias bound parameters and bandwidth

Usage

```
## S3 method for class 'bbnp_regression'
coef(object, ...)
```

Arguments

object	An object of class bbnp_regression
...	Additional arguments (unused)

Value

Named numeric vector with A (amplitude), r (decay rate), B (Y bound), and h (bandwidth)

Examples

```
X <- rnorm(100)
Y <- X^2 + rnorm(100)
fit <- biasBound_condExpectation(Y, X, h = 0.1)
coef(fit)
```

---

confint.bbnp_density *Extract Confidence Intervals from bbnp_density Object*

Description

Extracts confidence intervals for density estimates

Usage

```
## S3 method for class 'bbnp_density'
confint(object, parm = NULL, level = 0.95, ...)
```

Arguments

object	An object of class bbnp_density
parm	Not used (included for S3 generic compatibility)
level	Confidence level (default: 0.95). Note: this parameter is not used as the confidence level is fixed at object creation time.
...	Additional arguments (unused)

Value

For range estimation: a matrix with columns "lower" and "upper" For point estimation: a named vector with elements "lower" and "upper"

Examples

```
X <- rnorm(100)
fit <- biasBound_density(X, h = 0.1)
confint(fit)
```

confint.bbnp_regression

Extract Confidence Intervals from bbnp_regression Object

Description

Extracts confidence intervals for conditional expectation estimates

Usage

```
## S3 method for class 'bbnp_regression'
confint(object, parm = NULL, level = 0.95, ...)
```

Arguments

object	An object of class bbnp_regression
parm	Not used (included for S3 generic compatibility)
level	Confidence level (default: 0.95). Note: this parameter is not used as the confidence level is fixed at object creation time.
...	Additional arguments (unused)

Value

For range estimation: a matrix with columns "lower" and "upper" For point estimation: a named vector with elements "lower" and "upper"

Examples

```
X <- rnorm(100)
Y <- X^2 + rnorm(100)
fit <- biasBound_condExpectation(Y, X, h = 0.1)
confint(fit)
```

create_biasBound_config

Create a configuration object for bias bound estimations

Description

Create a configuration object for bias bound estimations

Usage

```
create_biasBound_config(
  X,
  Y = NULL,
  h = NULL,
  h_method = "cv",
  use_fft = TRUE,
  alpha = 0.05,
  resol = 100,
  xi_lb = NULL,
  xi_ub = NULL,
  methods_get_xi = "snr",
  noise_floor = "auto",
  envelope_use_Y = TRUE,
  integer_r = TRUE,
  kernel.fun = "Schennach2004",
  if_approx_kernel = TRUE,
  kernel.resol = 1000
)
```

Arguments

X	A numerical vector of sample data.
Y	Optional. A numerical vector of sample data for conditional expectation.
h	A scalar bandwidth parameter. If NULL, the bandwidth is automatically selected using the method specified in 'h_method'.
h_method	Method for automatic bandwidth selection when h is NULL. Options are "cv" (cross-validation) and "silverman" (Silverman's rule of thumb). Default is "cv".
use_fft	Ignored. Maintained for backward compatibility.
alpha	Confidence level for intervals.
resol	Resolution for the estimation range.
xi_lb	Lower bound for the interval of Fourier Transform frequency.
xi_ub	Upper bound for the interval of Fourier Transform frequency.
methods_get_xi	Method to determine the xi interval: "snr" (default), "Schennach", or "Schennach_loose".
noise_floor	Noise-floor form for the Schennach test: "auto" (default), "compact", or "general".
envelope_use_Y	If TRUE (default), fit the regression envelope to the cross-spectrum |phi_YX|; if FALSE, fit it to the marginal spectrum |phi_X|.
integer_r	If TRUE (default), when the fitted envelope slope falls below the minimum smoothness assumed by Schennach (2020, Definition 2), i.e. r < 2, clamp it up to r = 2 and refit A; this keeps the bias-bound integral finite. Slopes >= 2 are left unchanged.
kernel.fun	Kernel function to be used. Options include "normal", "epanechnikov", "Schennach2004", and "sinc".

`if_approx_kernel`
: Use approximations for the kernel function.

`kernel.resol`
: Resolution for kernel approximation.

Value

A configuration object (list) with all parameters

`create_kernel_functions` *Create kernel functions based on configuration*

Description

Create kernel functions based on configuration

Usage

```
create_kernel_functions(
  kernel.fun = "Schennach2004",
  if_approx_kernel = TRUE,
  kernel.resol = 1000
)
```

Arguments

`kernel.fun`
: A string specifying the kernel function to be used.

`if_approx_kernel`
: Logical. If TRUE, uses approximations for the kernel function.

`kernel.resol`
: The resolution for kernel function approximation.

Value

A list containing kernel function, its Fourier transform, and the kernel type

cv_bandwidth *Cross-Validation for Bandwidth Selection*

Description

Implements least-squares cross-validation for bandwidth selection with any kernel function. Uses FFT-based algorithm for n >= 100 (fast, O(m log m) complexity) and exact computation for n < 100 (accurate). The FFT method bins data onto a regular grid and computes the LSCV objective via convolution in the frequency domain.

Usage

```
cv_bandwidth(
  X,
  h_grid = NULL,
  kernel_func,
  kernel_type = "normal",
  grid_size = 512
)
```

Arguments

X	A numerical vector of sample data.
h_grid	A numerical vector of bandwidth values to evaluate. If NULL (default), a grid is automatically generated based on the range and distribution of the data.
kernel_func	The kernel function to use for cross-validation.
kernel_type	A string identifying the kernel type, used only for reference bandwidth.
grid_size	Number of grid points for FFT-based evaluation (used when n >= 100). Default is 512. Larger values increase accuracy but reduce speed. Automatically rounded up to the next power of 2.

Value

A scalar representing the optimal bandwidth that minimizes the cross-validation score.

Examples

```
# Generate sample data
X <- rnorm(100)
# Get optimal bandwidth using cross-validation with a normal kernel
kernel_functions <- create_kernel_functions("normal")
h_opt <- cv_bandwidth(X, kernel_func = kernel_functions$kernel,
                      kernel_type = kernel_functions$kernel_type)
```

fitted.bbnp_regression *Extract Fitted Values from bbnp_regression Object*

Description

Extracts the fitted conditional expectation values $E[Y|X = x]$

Usage

```
## S3 method for class 'bbnp_regression'
fitted(object, ...)
```

Arguments

object	An object of class bbnp_regression
...	Additional arguments (unused)

Value

Numeric vector of fitted values for range estimation, or a single numeric value for point estimation

Examples

```
X <- rnorm(100)
Y <- X^2 + rnorm(100)
fit <- biasBound_condExpectation(Y, X, h = 0.1)
fitted(fit)
```

gen_sample_data *Generate Sample Data*

Description

This function used for generate some sample data for experiment

Usage

```
gen_sample_data(size, dgp, seed = NULL)
```

Arguments

size	control the sample size.
dgp	data generating process, have options "normal", "chisq", "mixed", "poly", "2_fold_uniform".
seed	random seed number.

Value

A numeric vector of length `size`. The elements of the vector are generated according to the specified `dgp`:

normal Normally distributed values with mean 0 and standard deviation 2.

chisq Chi-squared distributed values with df = 10.

mixed Half normally distributed (mean 0, sd = 2) and half chi-squared distributed (df = 10) values.

poly Values from a polynomial cumulative distribution function on `[0,1]`.

2_fold_uniform Sum of two uniformly distributed random numbers.

Examples

```
# Generate 500 samples from 2-fold uniform distribution
X <- gen_sample_data(500, "2_fold_uniform", seed = 123)
```

---

plot.bbnp_density *Plot Method for bbnp_density Objects*

Description

Creates visualizations of bias-bounded density estimation results

Usage

```
## S3 method for class 'bbnp_density'
plot(
  x,
  type = c("density", "ft"),
  fill_ci = .bbnp_pal[["ci"]],
  fill_bias = .bbnp_pal[["bias"]],
  alpha_ci = 0.55,
  alpha_bias = 0.55,
  ft_resol = 500,
  xi_range = NULL,
  expand = 2.5,
  ...
)
```

Arguments

`x`	An object of class bbnp_density
`type`	Character string specifying plot type. Options are: • `"density"` (default): Density estimate with bias bounds and confidence intervals

- `"ft"`: Fourier transform plot with estimated envelope

`fill_ci`	Color for confidence interval ribbon (default: a muted blue).
`fill_bias`	Color for bias bound ribbon (default: a muted terracotta).
`alpha_ci`	Transparency for confidence interval ribbon (default: 0.30)
`alpha_bias`	Transparency for bias bound ribbon (default: 0.45)
`ft_resol`	Resolution for Fourier transform plot (default: 500)
`xi_range`	Optional numeric `c(lower, upper)` giving the frequency range to *display* in the `"ft"` plot. If `NULL` (default) a wide range around the selected window is shown (see `expand`). This controls only what is drawn; it does not change the fitting window `[xi_lb, xi_ub]`.
`expand`	For the `"ft"` plot when `xi_range` is `NULL`: how far past the selected window to display, as a multiple (default 2.5).
`...`	Additional arguments (unused)

Value

A ggplot2 object

Examples

```
X <- rnorm(100)
fit <- biasBound_density(X, h = 0.1)
plot(fit)
plot(fit, type = "ft")
```

`plot.bbnp_regression` *Plot Method for bbnp_regression Objects*

Description

Creates visualizations of bias-bounded conditional expectation estimation results

Usage

```
## S3 method for class 'bbnp_regression'
plot(
  x,
  type = c("regression", "ft"),
  fill_ci = .bbnp_pal[["ci"]],
  alpha_ci = 0.55,
  point_alpha = 0.28,
  point_color = .bbnp_pal[["point"]],
  ft_resol = 500,
  xi_range = NULL,
  expand = 2.5,
  ...
)
```

Arguments

x	An object of class bbnp_regression
type	Character string specifying plot type. Options are: • "regression" (default): Conditional expectation with confidence interval • "ft": Fourier transform plot with estimated envelope
fill_ci	Color for confidence interval ribbon (default: a muted blue).
alpha_ci	Transparency for confidence interval ribbon (default: 0.35)
point_alpha	Transparency for data points (default: 0.28)
point_color	Color for data points (default: a soft grey).
ft_resol	Resolution for Fourier transform plot (default: 500)
xi_range	Optional numeric c(lower, upper) giving the frequency range to *display* in the "ft" plot. If NULL (default) a wide range around the selected window is shown (see expand). This controls only what is drawn; it does not change the fitting window [xi_lb, xi_ub].
expand	For the "ft" plot when xi_range is NULL: how far past the selected window to display, as a multiple (default 2.5).
...	Additional arguments (unused)

Value

A ggplot2 object

Examples

```
X <- rnorm(100)
Y <- X^2 + rnorm(100)
fit <- biasBound_condExpectation(Y, X, h = 0.1)
plot(fit)
plot(fit, type = "ft")
```

---

print.bbnp_density *Print Method for bbnp_density Objects*

Description

Print Method for bbnp_density Objects

Usage

```
## S3 method for class 'bbnp_density'
print(x, digits = 4, ...)
```

Arguments

x	An object of class bbnp_density
digits	Number of digits to display (default: 4)
...	Additional arguments (unused)

Value

Invisibly returns the input object

---

print.bbnp_regression *Print Method for bbnp_regression Objects*

Description

Print Method for bbnp_regression Objects

Usage

```
## S3 method for class 'bbnp_regression'
print(x, digits = 4, ...)
```

Arguments

x	An object of class bbnp_regression
digits	Number of digits to display (default: 4)
...	Additional arguments (unused)

Value

Invisibly returns the input object

---

sample_data *Sample Data*

Description

Sample Data

Usage

```
sample_data
```

Format

A data frame with 1000 rows and 2 variables:

X Numeric vector, generated from 2 fold uniform distribution.

Y Numeric vector, `Y = -X^2 + 3*X + rnorm(1000)*X`.

`select_bandwidth` *Select Optimal Bandwidth*

Description

Selects an optimal bandwidth using the specified method.

Usage

```
select_bandwidth(
  X,
  Y = NULL,
  method = "cv",
  kernel.fun = "normal",
  if_approx_kernel = TRUE,
  kernel.resol = 1000
)
```

Arguments

`X`	A numerical vector of sample data.
`Y`	Optional. A numerical vector of sample data for conditional expectation estimation.
`method`	A string specifying the bandwidth selection method. Options are "cv" for cross-validation and "silverman" for Silverman's rule of thumb. Defaults to "cv".
`kernel.fun`	A string specifying the kernel type. Options include "normal", "epanechnikov", "Schennach2004", and "sinc".
`if_approx_kernel`	Logical. If TRUE, uses approximations for the kernel function.
`kernel.resol`	The resolution for kernel function approximation.

Value

A scalar representing the optimal bandwidth.

Examples

```
# Generate sample data
X <- rnorm(100)
# Get optimal bandwidth using cross-validation with normal kernel
h_opt <- select_bandwidth(X, method = "cv", kernel.fun = "normal")
# Get optimal bandwidth using Silverman's rule with Schennach kernel
h_opt <- select_bandwidth(X, method = "silverman", kernel.fun = "Schennach2004")
```

silverman_bandwidth *Silverman's Rule of Thumb for Bandwidth Selection*

Description

Implements Silverman's rule of thumb for selecting an optimal bandwidth in kernel density estimation.

Usage

```
silverman_bandwidth(X, kernel_type = "normal")
```

Arguments

X	A numerical vector of sample data.
kernel_type	A string identifying the kernel type.

Value

A scalar representing the optimal bandwidth.

Examples

```
# Generate sample data
X <- rnorm(100)
# Get optimal bandwidth using Silverman's rule
h_opt <- silverman_bandwidth(X, kernel_type = "normal")
```

summary.bbnp_density *Summary Method for bbnp_density Objects*

Description

Summary Method for bbnp_density Objects

Usage

```
## S3 method for class 'bbnp_density'
summary(object, ...)
```

Arguments

object	An object of class bbnp_density
...	Additional arguments (unused)

Value

An object of class summary.bbnp_density

summary.bbnp_regression *Summary Method for bbnp_regression Objects*

Description

Summary Method for bbnp_regression Objects

Usage

```
## S3 method for class 'bbnp_regression'
summary(object, ...)
```

Arguments

object	An object of class bbnp_regression
...	Additional arguments (unused)

Value

An object of class summary.bbnp_regression

Index