

# Package ‘baggingbwsel’

July 12, 2021

**Type** Package

**Title** Bagging Bandwidth Selection in Kernel Density and Regression Estimation

**Version** 1.0

**Date** 2021-07-08

**Description** Bagging bandwidth selection methods for the Parzen-Rosenblatt and Nadaraya-Watson estimators. These bandwidth selectors can achieve greater statistical precision than their non-bagged counterparts while being computationally fast. See Barreiro-Ures et al. (2020) <[doi:10.1093/biomet/asaa092](https://doi.org/10.1093/biomet/asaa092)> and Barreiro-Ures et al. (2021) <[arXiv:2105.04134](https://arxiv.org/abs/2105.04134)>.

**License** GPL-3

**Encoding** UTF-8

**Depends** mclust, foreach

**Imports** Rcpp (>= 1.0.3), parallel, doParallel, kedd, stats, sm, nor1mix, rgl, rpanel, tkrplot, misc3d

**LinkingTo** Rcpp

**RoxygenNote** 7.1.1

**NeedsCompilation** yes

**Author** Daniel Barreiro-Ures [aut, cre],  
Jeffrey Hart [aut],  
Ricardo Cao [aut],  
Mario Francisco-Fernandez [aut]

**Maintainer** Daniel Barreiro-Ures <[daniel.barreiro.ures@udc.es](mailto:daniel.barreiro.ures@udc.es)>

**Repository** CRAN

**Date/Publication** 2021-07-12 07:20:02 UTC

## R topics documented:

baggingbwsel-package	2
bagcv	3

bagreg . . . . .	4
hboot_bag . . . . .	5
hsss_dens . . . . .	6
mopt . . . . .	7
tss_dens . . . . .	8
<b>Index</b>	<b>9</b>

---

baggingbwsel-package *A short title line describing what the package does*

---

## Description

A more detailed description of what the package does. A length of about one to five lines is recommended.

## Details

This section should provide a more detailed overview of how to use the package, including the most important functions.

## Author(s)

Your Name, email optional.

Maintainer: Your Name <your@email.com>

## References

This optional section can contain literature or other references for background information.

## See Also

Optional links to other man pages

## Examples

```
## Not run:
## Optional simple examples of the most important functions
## These can be in \dontrun{} and \donttest{} blocks.

## End(Not run)
```

---

bagcv *Bagged CV bandwidth selector for Parzen-Rosenblatt estimator*

---

## Description

Bagged CV bandwidth selector for Parzen-Rosenblatt estimator

## Usage

```
bagcv(x, r, s, h0, h1, nb = r, ncores = parallel::detectCores())
```

## Arguments

x	Vector. Sample.
r	Positive integer. Size of the subsamples.
s	Positive integer. Number of subsamples.
h0	Positive real number. Range over which to minimize, left bound.
h1	Positive real number. Range over which to minimize, right bound.
nb	Positive integer. Number of bins.
ncores	Positive integer. Number of cores with which to parallelize the computations.

## Details

Bagged cross-validation bandwidth selector for the Parzen-Rosenblatt estimator.

## Value

Bagged CV bandwidth.

## Examples

```
set.seed(1)
x <- rnorm(10^6)
bagcv(x, 5000, 100, 0.01, 1, 1000, 2)
```

---

**bagreg***Bagged CV bandwidth selector for Nadaraya-Watson estimator*

---

**Description**

Bagged CV bandwidth selector for Nadaraya-Watson estimator

**Usage**

```
bagreg(x, y, r, s, h0, h1, nb = r, ncores = parallel::detectCores())
```

**Arguments**

x	Covariate vector.
y	Response vector.
r	Positive integer. Size of the subsamples.
s	Positive integer. Number of subsamples.
h0	Positive real number. Range over which to minimize, left bound.
h1	Positive real number. Range over which to minimize, right bound.
nb	Positive integer. Number of bins to use in cross-validation.
ncores	Positive integer. Number of cores with which to parallelize the computations.

**Details**

Bagged cross-validation bandwidth selector for the Nadaraya-Watson estimator.

**Value**

Bagged CV bandwidth.

**Examples**

```
set.seed(1)
x <- rnorm(10^5)
y <- 2*x+rnorm(1e5,0,0.5)
bagreg(x, y, 1000, 10, 0.01, 1, 1000, 2)
```

---

hboot_bag	<i>Bagging bootstrap bandwidth selector for Parzen-Rosenblatt estimator</i>
-----------	---

---

## Description

Bagging bootstrap bandwidth selector for Parzen-Rosenblatt estimator

## Usage

```
hboot_bag(  
  x,  
  m = n,  
  N = 1,  
  nb = 1000L,  
  g,  
  lower,  
  upper,  
  ncores = parallel::detectCores(logical = FALSE)  
)
```

## Arguments

x	Vector. Sample.
m	Positive integer. Size of the subsamples.
N	Positive integer. Number of subsamples.
nb	Positive integer. Number of bins.
g	Positive real number. Pilot bandwidth.
lower	Positive real number. Range over which to minimize, left bound.
upper	Positive real number. Range over which to minimize, right bound.
ncores	Positive integer. Number of cores with which to parallelize the computations.

## Details

Bagging bootstrap bandwidth selector for the Parzen-Rosenblatt estimator.

## Value

Bagged CV bandwidth.

## Examples

```
set.seed(1)  
x <- rnorm(10^5)  
hboot_bag(x, 5000, 10, 1000, lower=0.001, upper=1, ncores=2)
```

---

hsss_dens	<i>Generalized bagging CV bandwidth selector for Parzen-Rosenblatt estimator</i>
-----------	--

---

## Description

Generalized bagging CV bandwidth selector for Parzen-Rosenblatt estimator

## Usage

```
hsss_dens(x, r, s, nb = r, h0, h1, ncores = parallel::detectCores())
```

## Arguments

x	Vector. Sample.
r	Positive integer. Size of the subsamples.
s	Positive integer. Number of subsamples.
nb	Positive integer. Number of bins.
h0	Positive real number. Range over which to minimize, left bound.
h1	Positive real number. Range over which to minimize, right bound.
ncores	Positive integer. Number of cores with which to parallelize the computations.

## Details

Generalized bagging cross-validation bandwidth selector for the Parzen-Rosenblatt estimator.

## Value

Bagged CV bandwidth.

## Examples

```
set.seed(1)
x <- rnorm(10^5)
hsss_dens(x, 5000, 100, 1000, 0.001, 1, 2)
```

---

mopt	<i>Estimation of the optimal subsample size for bagged CV bandwidth for Parzen-Rosenblatt estimator</i>
------	---

---

## Description

Estimation of the optimal subsample size for bagged CV bandwidth for Parzen-Rosenblatt estimator

## Usage

```
mopt(x, N, r = 1000, s = 100, ncores = parallel::detectCores())
```

## Arguments

x	Vector. Sample.
N	Positive integer. Number of subsamples for the bagged bandwidth.
r	Positive integer. Size of the subsamples.
s	Positive integer. Number of subsamples.
ncores	Positive integer. Number of cores with which to parallelize the computations.

## Details

Estimates the optimal size of the subsamples for the bagged CV bandwidth selector for the Parzen-Rosenblatt estimator.

## Value

Estimate of the optimal subsample size.

## Examples

```
set.seed(1)
x <- rt(10^5, 5)
mopt(x, 500, 500, 10, 2)
```

---

tss_dens	<i>Second order bagging CV bandwidth selector for Parzen-Rosenblatt estimator</i>
----------	---

---

**Description**

Second order bagging CV bandwidth selector for Parzen-Rosenblatt estimator

**Usage**

```
tss_dens(x, r, s, h0, h1, nb = 1000, ncores = 1)
```

**Arguments**

x	Vector. Sample.
r	Vector. The two subsample sizes.
s	Positive integer. Number of subsamples.
h0	Positive real number. Range over which to minimize, left bound.
h1	Positive real number. Range over which to minimize, right bound.
nb	Positive integer. Number of bins.
ncores	Positive integer. Number of cores with which to parallelize the computations.

**Details**

Second order bagging cross-validation bandwidth selector for the Parzen-Rosenblatt estimator.

**Value**

Second order bagging CV bandwidth.

**Examples**

```
set.seed(1)
x <- rnorm(10^5)
tss_dens(x, 5000, 10, 0.01, 1, 1000, 2)
```



# Index

**\* package**

baggingbwsel-package, [2](#)

bagcv, [3](#)

baggingbwsel (baggingbwsel-package), [2](#)

baggingbwsel-package, [2](#)

bagreg, [4](#)

hboot\_bag, [5](#)

hsss\_dens, [6](#)

mopt, [7](#)

tss\_dens, [8](#)