Package 'baggingbwsel'

Type Package

Title Bagging Bandwidth Selection in Kernel Density and Regression
Estimation

Version 1.1

Date 2024-07-22

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> and
Barreiro-Ures et al. (2021) <doi:10.48550/arXiv.2105.04134>.

License GPL-3

URL https://rubenfcasal.github.io/baggingbwsel/,
https://github.com/rubenfcasal/baggingbwsel/

BugReports https://github.com/rubenfcasal/baggingbwsel/issues/

Encoding UTF-8

Depends mclust, foreach

Imports Rcpp (>= 1.0.3), parallel, doParallel, kedd, stats, sm,
nor1mix, misc3d

Suggests rgl, tkrplot, rpanel

LinkingTo Rcpp

RoxygenNote 7.3.2

NeedsCompilation yes

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

Maintainer Ruben Fernandez-Casal <rubenfcasal@gmail.com>

Repository CRAN

Date/Publication 2024-07-27 16:20:09 UTC

Contents

---

baggingbwsel-package	*baggingbwsel: Bagging bandwidth selection in kernel density and re-gression estimation*

Description

This package implements bagging bandwidth selection methods for the Parzen-Rosenblatt kernel density estimator, and for the Nadaraya-Watson and local polynomial kernel regression estimators. These bandwidth selectors can achieve greater statistical precision than their non-bagged counterparts while being computationally fast. See Barreiro-Ures et al. (2021a) and Barreiro-Ures et al. (2021b).

Author(s)

Maintainer: Ruben Fernandez-Casal <rubenfcasal@gmail.com>

Authors:

- Daniel Barreiro-Ures <daniel.barreiro.ures@udc.es>

- Jeffrey Hart

- Ricardo Cao

- Mario Francisco-Fernandez

References

Barreiro-Ures, D., Cao, R., Francisco-Fernández, M., & Hart, J. D. (2021a). Bagging cross-validated bandwidths with application to big data. *Biometrika*, **108**(4), 981-988, doi:10.1093/biomet/asaa092.

Barreiro-Ures, D., Cao, R., & Francisco-Fernández, M. (2021b). Bagging cross-validated bandwidth selection in nonparametric regression estimation with applications to large-sized samples. *arXiv preprint*, doi:10.48550/arXiv.2105.04134.

See Also

Useful links:

- <https://rubenfcasal.github.io/baggingbwsel/>
- <https://github.com/rubenfcasal/baggingbwsel/>
- Report bugs at <https://github.com/rubenfcasal/baggingbwsel/issues/>

---

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 local polynomial kernel regression.*

Description

Bagged CV bandwidth selector for local polynomial kernel regression.

Usage

```
bagreg(
  x,
  y,
  r,
  s,
  h0,
  h1,
  nb = r,
  ncores = parallel::detectCores(),
  poly.index = 0
)
```

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.
poly.index	Non-negative integer defining local constant (0) or local linear (1) smoothing. Default value: 0 (Nadaraya-Watson estimator).

Details

Bagged cross-validation bandwidth selector for local polynomial kernel regression.

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	*Bagging bootstrap bandwidth selector for Parzen-Rosenblatt estimator*

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	*Generalized bagging CV bandwidth selector for Parzen-Rosenblatt es-timator*

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

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	*Second order bagging CV bandwidth selector for Parzen-Rosenblatt estimator*

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