Package ‘sBF’

February 20, 2015

Type Package
Title Smooth Backfitting
Version 1.1.1
Date 2014-12-19
Author A. Arcagni, L. Bagnato
Maintainer Alberto Arcagni <a.arcagni@campus.unimib.it>
Description Smooth Backfitting for additive models using
    Nadaraya-Watson estimator
License GPL (>= 2)
LazyLoad yes
Repository CRAN
Date/Publication 2014-12-19 17:37:38
NeedsCompilation no

R topics documented:

sBF-package .................................................. 1
K .............................................................. 2
sBF ............................................................ 3

Index

sBF-package Smooth Backfitting Estimator Package

Description

Smooth Backfitting Estimator

Details
Author(s)
A. Arcagni a.arcagni@campus.unimib.it and L. Bagnato luca.bagnato@unimib.it

References

See Also
sBF, K.

**K**

Kernel weighting function

Description
Instrumental to the sBF function. It returns weights used in the Nadaraya-Watson estimator.

Usage

```
K(u, method = "gaussian")
```

Arguments

- `u` : distance from the origin.
- `method` : type of kernel function. The default value is gaussian, other possible methods are: uniform, epanechnikov, biweight, and triweight.

Details
The domain of the kernel functions is centered at the origin and generally the weight value returned by the kernel decreases while the distance `u` from the origin increases.
References

See Also
`sbf-package`, `sbf`.

`sbf` Smooth Backfitting Estimator

Description
Smooth Backfitting for additive models using Nadaraya-Watson estimator.

Usage
`sbf(dat, depCol = 1, m = 100, windows = rep(20, ncol(dat) - 1),
bw = NULL, method = "gaussian", mx = 100, epsilon = 1e-04,
PP = NULL, G = NULL)`

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dat</td>
<td>matrix of data.</td>
</tr>
<tr>
<td>depCol</td>
<td>column of dat matrix in which the dependent variable is positioned.</td>
</tr>
<tr>
<td>m</td>
<td>number of grid points. Higher values of m imply better estimates and longer computational time.</td>
</tr>
<tr>
<td>windows</td>
<td>number of windows. (covariate range width)/windows provide the bandwidths for the kernel regression smoother.</td>
</tr>
<tr>
<td>bw</td>
<td>bandwidths for the kernel regression smoother.</td>
</tr>
<tr>
<td>method</td>
<td>kernel method. See function <code>K</code>.</td>
</tr>
<tr>
<td>mx</td>
<td>maximum iterations number.</td>
</tr>
<tr>
<td>epsilon</td>
<td>convergence limit of the iterative algorithm.</td>
</tr>
<tr>
<td>PP</td>
<td>matrix of joint probabilities.</td>
</tr>
<tr>
<td>G</td>
<td>grid on which univariate functions are estimated.</td>
</tr>
</tbody>
</table>

Details
Bandwidth can be chosen in two different ways: through the argument `bw` or defining the number of windows into the range of the values of any independent variable through the argument `windows` (equal to 20 by default). Bandwidth is the width of the windows. Both the parameters `bw` and `windows` can be single values, then every smoother has the same bandwidth, or they can be vectors of length equal to the covariates number to specify different bandwidths for any direction. Higher values of the bandwidth provide smoother estimates.
In applications it could be useful using the same PP matrix for different estimates, e.g. to evaluate the impact of different bandwidths and develop algorithms to select optimal bandwidths (see, for example Nielsen and Sperlich, 2005, page 52). This reasoning applies also to the grid G. This is why the possibility to input matrices G and PP as parameters is given. The program creates G and PP if they are not inserted.

Value

<table>
<thead>
<tr>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>mxhat</td>
</tr>
<tr>
<td>m0</td>
</tr>
<tr>
<td>grid</td>
</tr>
<tr>
<td>conv</td>
</tr>
<tr>
<td>nit</td>
</tr>
<tr>
<td>PP</td>
</tr>
<tr>
<td>bw</td>
</tr>
</tbody>
</table>

See Also

sBF-package, K.

Examples

```r
X <- matrix(rnorm(1000), ncol=2)
MX1 <- X[,1]^3
MX2 <- sin(X[,2])
Y <- MX1 + MX2
data <- cbind(Y, X)
est <- sBF(data)

par(mfrow=c(1, 2))
plot(est$grid[,1], est$mxhat[,1], type="l",
     ylab=expression(m[1](x[1])), xlab=expression(x[1]))
curve(x^3, add=TRUE, col="red")
plot(est$grid[,2], est$mxhat[,2], type="l",
     ylab=expression(m[2](x[2])), xlab=expression(x[2]))
curve(sin(x), add=TRUE, col="red")
par(mfrow=c(1, 1))
```


Index

K, 2, 2, 3, 4

sBF, 2, 3, 3
sBF-package, 1