Package ‘EDR’

September 14, 2016

Version 0.6-6
Date 2016-09-06
Title Estimation of the Effective Dimension Reduction (‘EDR’) Space
Author Joerg Polzehl <joerg.polzehl@wias-berlin.de>
Maintainer Joerg Polzehl <joerg.polzehl@wias-berlin.de>
Depends R (>= 2.14.0), sm
Suggests akima
Description The library contains R-functions to estimate the effective dimension reduction space in ‘multi-index’ regression models.
License GPL (>= 2)
Copyright This library is Copyright (C) 2006-2016 Weierstrass Institute for Applied Analysis and Stochastics.
URL http://www.wias-berlin.de/
Note This software comes with NO warranty!
NeedsCompilation yes
Repository CRAN
Date/Publication 2016-09-14 19:44:08

R topics documented:

<table>
<thead>
<tr>
<th>edr</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>edr.R</td>
<td>4</td>
</tr>
<tr>
<td>edrcv</td>
<td>5</td>
</tr>
<tr>
<td>plot.edr</td>
<td>8</td>
</tr>
<tr>
<td>predict.edr</td>
<td>9</td>
</tr>
<tr>
<td>print.edr</td>
<td>10</td>
</tr>
<tr>
<td>summary.edr</td>
<td>11</td>
</tr>
</tbody>
</table>

Index 14
### edr

*Estimation of the effective dimension reduction (EDR) space: Structure adaptive approach for dimension reduction*

#### Description

This function implements the algorithms, proposed in M. Hristache, A. Juditsky, J. Polzehl and V. Spokoiny (2001) and ... (2006), for estimation of the effective dimension reduction (EDR) space in multi-index regression models

\[ y = f(x) + \varepsilon = g(B_m^T x) + \varepsilon. \]

#### Usage

```r
edr(x, y, m = 2, rho0 = 1, h0 = NULL, ch = exp(0.5/max(4, (dim(x)[2]))),
    crhomin = 1, cm = 4, method = "Penalized", basis = "Quadratic", cw = NULL,
    graph = FALSE, show = 1, trace = FALSE, fx = NULL, R = NULL)
```

#### Arguments

- `x`: x specifies the design matrix, dimension \((n, d)\)
- `y`: y specifies the response, length \(n\).
- `m`: Rank of matrix M in case of `method="Penalized"`, not used for the other methods.
- `rho0`: Initial value for the regularization parameter \(\rho\).
- `h0`: Initial bandwidth.
- `ch`: Factor for decreasing \(h\) with iterations.
- `crhomin`: Factor to increase the default value of rhomin. This is just added to explore properties of the algorithms. Defaults to 1.
- `cm`: Factor in the definition of \(\Pi_k = C_m \rho_k^2 I_L + \hat{N}_{k-1}\). Only used if `method="Penalized"`.
- `method`: Specifies the algorithm to use. The default method="Penalized" corresponds to the algorithm proposed in ... (2006). method="HJPS" corresponds to the original algorithm from Hristache et.al. (2001) while method="HJPS2" specifies a modification (correction) of this algorithm.
- `basis`: Specifies the set of basis functions. Options are basis="Quadratic" (default) and basis="Linear".
- `cw`: cw another regularization parameter, secures identifiability of a minimum number of local gradient directions. Defaults to \(\frac{Q}{d}\). Has to be positive or NULL.
- `graph`: If `graph=TRUE` intermediate results are plotted.
- `show`: If `graph=TRUE` the parameter `show` determines the dimension of the EDR that is to be used when plotting intermediate results. If `trace=TRUE` and `is.null(R)` it determines the dimension of the EDR when computing the risk values.
trace=TRUE additional diagnostics are provided for each iteration. This includes current, at iteration \( k \), values of the regularization parameter \( \rho_k \) and bandwidth \( h_k \), normalized cumulative sums of eigenvalues of \( \hat{B} \) and if \( !is.null(R) \) two distances between the true, specified in \( R \) and estimated EDR.

**fx**

True values of \( f(x) \). This is just added to explore properties of the algorithms and not used in the algorithms.

**R**

True matrix \( R \). This is just added to explore properties of the algorithms and not used in the algorithms.

**Details**

See reference for details.

**Value**

Object of class "edr" with components.

- **x** The design matrix.
- **y** The values of the response.
- **bhat** Matrix \( \hat{B} \) characterizing the effective dimension space. For a specified dimension \( m \), \( \hat{B}_m = BO_m \), with \( \hat{B}^T \hat{B} = \Omega \Lambda \Omega^T \) being the eigenvalue decomposition of \( \hat{B}^T \hat{B} \), specifies the projection to the \( m \)-dimensional subspace that provides the best approximation.
- **fhat** an highly oversmoothed estimate of the values of the regression function at the design points. This is provided as a backup only for the case that package \( sm \) is not installed.
- **cumlam** Cummulative amount of information explained by the first components of \( \hat{B} \).
- **nmean** Mean numbers of observations used in each iteration.
- **h** Final bandwidth
- **rho** Final value of \( \rho \)
- **h0** Initial bandwidth
- **rho0** Initial value of \( \rho \)
- **cm** The factor \( cm \)
- **call** Arguments of the call to edr

**Author(s)**

Joerg Polzehl, <polzehl@wias-berlin.de>

**References**

Eigenvectors of the effective dimension reduction (EDR) space

Description

Computes the eigenvectors of the effective dimension reduction (EDR) space obtained by function edr.

Usage

edr.R(B, m)

Arguments

B Either an object of class edr created by edr or the list component bhat of such an object.
m Dimension of the effective dimension reduction (EDR) space. m=1 corresponds to single index models, m>1 specifies a multiindex model.

Value

Matrix of dimension c(m, d) containing the m eigenvectors as rows.

Author(s)

Joerg Polzehl, <polzehl@wias-berlin.de>

References


See Also

edr

Examples

require(EDR)
demo(edr_ex1)
demo(edr_ex2)
**Examples**

```r
require(EDR)
## Not run: demo(edr_ex1)
## Not run: demo(edr_ex2)
```

---

**edrcv**

*Risk assessment by Cross-Validation*

**Description**

This function, additionally to estimating the effective dimension reduction space (EDR), see also function `edr`, estimates the Mean Squared Error of Prediction (MSEP) and the Mean Absolute Error of Prediction (MAEP) when using the estimated EDR by Cross-Validation. Estimates of the regression function are produced using function `sm.regression` from package `sm`.

**Usage**

```r
edrcv(x, y, m = 2, rho0 = 1, h0 = NULL, ch = exp(0.5/max(4, (dim(x)[2]))), crhomin = 1, cm = 4, method = "Penalized", fit = "sm", basis = "Quadratic", cw = NULL, graph = FALSE, show = 1, trace = FALSE, seed = 1, cvsize = 1, m0 = min(m, 2), hsm = NULL)
```

**Arguments**

- `x` x specifies the design matrix, dimension `(n,d)`
- `y` y specifies the response, length `n`.
- `m` Rank of matrix M in case of method="Penalized", not used for the other methods.
- `rho0` Initial value for the regularization parameter \( \rho \).
- `h0` Initial bandwidth.
- `ch` Factor for decreasing \( h \) with iterations.
- `crhomin` Factor to increase the default value of rhomin. This is just added to explore properties of the algorithms. Defaults to 1.
- `cm` Factor in the definition of \( \Pi_k = C_m \rho_k^2 I_L + \tilde{M}_{k-1} \). Only used if method="Penalized".
- `method` Specifies the algorithm to use. The default method="Penalized" corresponds to the algorithm proposed in ... (2006). method="HJPS" corresponds to the original algorithm from Hristache et.al. (2001) while method="HJPS2" specifies a modification (correction) of this algorithm.
- `fit` Specifies the method for estimating and predicting values of the link function. This can either be `fit="sm"` specifying use of the sm package or `fit="direct"` specifying the use of a local linear smoother. In case of \( m0>2 \) `fit="direct"` is used due to restrictions in the sm package.
- `basis` Specifies the set of basis functions. Options are `basis="Quadratic"` (default) and `basis="Linear"`. 
cw another regularization parameter, secures identifiability of a minimum number of local gradient directions. Defaults to 1/d . Has to be positive or NULL.

graph If graph==TRUE intermediate results are plotted.

show If graph==TRUE the parameter show determines the dimension of the EDR that is to be used when plotting intermediate results. If trace==TRUE and !is.null(R) it determines the dimension of the EDR when computing the risk values.

trace trace==TRUE additional diagnostics are provided for each iteration. This includes current, at iteration k, values of the regularization parameter \( \rho_k \) and bandwidth \( h_k \), normalized cimmulative sums of eigenvalues of \( \hat{B} \) and if !is.null(R) two distances between the true, specified in \( R \) and estimated EDR.

seed Seed for generating random groups for CV

cvsize Groupsize k in leave-k-out CV

\( m0 \) Dimension of the dimension reduction space to use when fitting the data. Should be either 1 or 2.

hsm If is.null(hsm) the bandwidth used by sm.regression for smoothing within the EDR is chosen by cross-validation within sm.regression when needed. Alternatively a grid of bandwidths may be specified. In that case a bandwidth for sm.regression is chosen from the grid that minimizes the estimated mean absolute error of prediction.

Details

This function performs a leave-k-out cross-validation to estimate the risk in terms of Mean Squared Error of Prediction (MSEP) and Mean Absolute Error of Prediction (MAEP) when using function edr to estimate an effective dimension reduction space of dimension \( m0 \) and using this estimated space to predict values of the response. Smoothing within the dimension reduction space is performed using the function sm.regression from package sm. The bandwidth for sm.regression is chosen by Cross-Validation.

Value

Object of class "edr" with components.

\( x \) The design matrix.

\( y \) The values of the response.

\( \hat{b} \) Matrix \( \hat{B} \) characterizing the effective dimension space. For a specified dimension \( m \) \( \hat{B}_m = \hat{B}O_m \), with \( \hat{B}^T \hat{B} = O \Lambda O^T \) being the eigenvalue decomposition of \( \hat{B}^T \hat{B} \), specifies the projection to the \( m \)-dimensional subspace that provides the best approximation.

\( \hat{f} \) an highly oversmoothed estimate of the values of the regression function at the design points. This is provided as a backup only for the case that package sm is not installed.

\( cumlam \) Cumulative amount of information explained by the first components of \( \hat{B} \).

\( nmean \) Mean numbers of observations used in each iteration.

\( h \) Final bandwidth
edrcv

rho Final value of $\rho$

h0 Initial bandwidth

rho0 Initial value of $\rho$

cm The factor cm

call Arguments of the call to edrcv

cvmres Residuals from cross-validation.

cvmseofh Estimates of MSEP for bandwidths hsm

cvmaeofh Estimates of MAEP for bandwidths hsm

cvmse Estimate of MSEP

cvmae Estimate of MAEP

hsm Set of bandwidths specified for use with sm.regression

hsmopt Bandwidth selected for use with sm.regression if hsm was specified.

**Note**

This function requires package sm if fit="sm".

**Author(s)**

Joerg Polzehl, <polzehl@wias-berlin.de>

**References**


**See Also**

edr.plot.edr, summary.edr, print.edr, edr.R, predict.edr

**Examples**

```r
require(EDR)
# Not run: demo(edr_ex4)
```
plot.edr

Plot results produced by function edr.

Description

Illustrate the fitted model within the estimated one or two-dimensional effective dimension reduction (EDR) space.

Usage

## S3 method for class 'edr'
plot(x, m = 1, ylab = "Y", title = ", sm = TRUE, ...)

Arguments

x 
Object of class "edr".

m 
Dimension of the effective dimension reduction (EDR) space. m=1 corresponds to single index models, m>1 specifies a multiindex model. Values of m>2 are currently not allowed.

ylab 
Label for the response variable.

title 
Optional title.

sm 
If sm==TRUE nonparametric regression is performed within the m-dimensional EDR using function sm.regression from package sm. If sm==FALSE oversmoothed fitted values calculated within function edr are used for visualisation.

... 
Additional parameters will be passed to plot in case of m=1 and to image if m=2.

Value

Returns invisible(NULL).

Author(s)

Joerg Polzehl, <polzehl@wias-berlin.de>

References


See Also

edr, edr.R, print.edr, summary.edr
Examples

```r
require(EDR)
## Not run: demo(edr_ex1)
## Not run: demo(edr_ex2)
```

### predict.edr

**Predict values of the link function for a fitted edr-object**

**Description**

The function allows to predict values of the link function in a multi-index model with estimated effective dimension reduction space.

**Usage**

```r
## S3 method for class 'edr'
predict(object, xest, m = 1, h=NULL, method = "sm",...)
```

**Arguments**

- `object`: object of class `edr`
- `xest`: matrix of design points where values of the link function are to be predicted
- `m`: specified dimension of the dimension reduction space.
- `h`: bandwidth, if `h=NULL` a bandwidth is guessed from the estimation
- `method`: method=“sm” requires and uses package sm while method=“direct” specifies a local linear smoother.
- `...`: Additional parameters that are currently not evaluated.

**Details**

This function provides predictions of response values based on a multi-index model analyzed using function edr. It requires specification of the dimension of the dimension reduction space in argument `m`. Argument `method` allows to select between two different local smoothers for estimation of values of the link function. The use of `method = "sm"` requires package sm and is restricted to `m<2`, see documentation of package sm. `method = "direct"` uses an implementation of a local linear smoother. If parameters of the local linear smoother are not identified for a given bandwidth `h` a kernel smoother or if this is not identified a 1-nearest neighbor estimate is used. The bandwidth is, in case of `h=NULL`, is determined by generalized cross-validation for sm and as $x$h$x$rho in case of the local linear smoother.

**Value**

List with components.

- `x`: The content of argument `xest`
- `fhat`: Predicted values of the link function
Note
This function requires package sm if method="sm".

Author(s)
Joerg Polzehl, <polzehl@wias-berlin.de>

References
Statistical Computation and Simulation*, DOI: 10.1080/00949650801959699

See Also
edr.plot.edr, summary.edr, print.edr, edr.R, edrcv

Examples
```r
require(EDR)
## Not run: demo(edr_ex4)
```

```
print.edr
  Print information for objects produced by function edr.
```

Description
The function provides information on the estimated effective dimension reduction (EDR) space.

Usage
```r
## S3 method for class 'edr'
print(x, m = 1, R = NULL, ...)
```

Arguments
- `x` Object of class "edr".
- `m` Dimension of the effective dimension reduction (EDR) space. \( m = 1 \) corresponds
to single index models, \( m > 1 \) specifies a multiindex model. Determines the num-
ber of eigenvectors and cumulative eigenvalues to show.
- `R` If code \( R \) specifies a matrix (dimension \( c(k, d), k \geq m, d=dim(x \times x)[2] \), this
  matrix is interpreted as spanning the true EDR space. Two distances between
  the estimated EDR space and the space spanned \( R[1:m,] \) are computed.
- `...` Additional parameters will be ignored
Details

Provides information on the estimated effective dimension reduction (EDR) space. The first $m$ basis vectors and the cumulative sum of normalized eigenvalues of matrix $\text{objectDbhat}$ are given. If $R$ is specified the distance

$$||R(I - \hat{P}_m)||/||R||$$

and the distance specified by Li (1992) are computed.

Value

Returns invisible(NULL).

Author(s)

Joerg Polzehl,<polzehl@wias-berlin.de>

References


See Also

edr, edr.R, summary.edr, plot.edr

Examples

```
require(EDR)
## Not run: demo(edr_ex1)
## Not run: demo(edr_ex2)
```
Arguments

object Object of class "edr".

m Dimension of the effective dimension reduction (EDR) space. m=1 corresponds to single index models, m>1 specifies a multiindex model. Determines the number of eigenvectors and cumulative eigenvalues to show.

R If code R specifies a matrix (dimension c(k,d), k \geq m, d=dim(object$x)[2], this matrix is interpreted as spanning the true EDR space. Two distances between the estimated EDR space and the space spanned R[1:m,] are computed.

... Additional parameters will be ignored

Details

Provides information on the estimated effective dimension reduction (EDR) space. The first m basis vectors and the cumulative sum of normalized eigenvalues of matrix object$bhat$ are given. If R is specified the distance

\[
\| R(I - \hat{P}_m) \| / \| R \| \text{ with } \hat{P}_m = U_m^T U_m, \hat{R}_m = U_m \Lambda V^T
\]

and the distance specified by Li (1992) are computed.

Value

Returns a list with components

Rhat (First) m eigenvectors of the estimated EDR space.
cumlam Cumulative sum of first m eigenvalues of object$bhat$, standardized by the sum of all eigenvalues.

loss1 If R was specified the distance \(\| R(I - \hat{P}_m) \| / \| R \| \) between the true and estimated m-dimensional EDR space.

loss2 The distance specified by Li (1992).

Author(s)

Joerg Polzehl, <polzehl@wias-berlin.de>

References


See Also

edr, edr.R, print.edr, plot.edr
Examples

```r
require(EDR)
## Not run: demo(edr_ex1)
## Not run: demo(edr_ex2)
```
Index

*Topic **hplot**
  
  plot.edr, 8

*Topic **regression**

  edr, 2
  
edr.R, 4
  edrcv, 5
  plot.edr, 8
  predict.edr, 9
  print.edr, 10
  summary.edr, 11

*Topic **smooth**

  edr, 2
  edrcv, 5
  predict.edr, 9
  print.edr, 10
  summary.edr, 11

edr, 2, 4, 5, 7, 8, 10–12
edr.R, 4, 4, 7, 8, 10–12
edrcv, 4, 5, 10

plot.edr, 4, 7, 8, 10–12
predict.edr, 7, 9
print.edr, 4, 7, 8, 10, 10, 12

summary.edr, 4, 7, 8, 10, 11, 11