# Package 'hfr'

February 27, 2024

**Type** Package

**Title** Estimate Hierarchical Feature Regression Models

**Date** 2024-02-27

**Version** 0.7.1

**Author** Johann Pfitzinger [aut, cre]

**Maintainer** Johann Pfitzinger <johann.pfitzinger@gmail.com>

**Description** Provides functions for the estimation, plotting, predicting and cross-validation of hierarchical feature regression models as described in Pfitzinger (2024). Cluster Regularization via a Hierarchical Feature Regression. Econometrics and Statistics (in press). <doi:10.1016/j.ecosta.2024.01.003>.

**License** GPL-2

**Imports** quadprog, stats, dendextend, RColorBrewer, corpcor

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**URL** https://hfr.residualmetrics.com,
https://github.com/jpfitzinger/hfr

**Suggests** MASS, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2024-02-27 19:40:06 UTC

## R topics documented:

## `cv.hfr` *Cross validation for a hierarchical feature regression*

### Description

HFR is a regularized regression estimator that decomposes a least squares regression along a supervised hierarchical graph, and shrinks the edges of the estimated graph to regularize parameters. The algorithm leads to group shrinkage in the regression parameters and a reduction in the effective model degrees of freedom.

### Usage

```
cv.hfr(
  x,
  y,
  weights = NULL,
  kappa = seq(0, 1, by = 0.1),
  q = NULL,
  intercept = TRUE,
  standardize = TRUE,
  nfolds = 10,
  foldid = NULL,
  partial_method = c("pairwise", "shrinkage"),
  l2_penalty = 0,
  ...
)
```

### Arguments

| | |
|---|---|
| `x` | Input matrix or data.frame, of dimension $(N \times p)$; each row is an observation vector. |
| `y` | Response variable. |
| `weights` | an optional vector of weights to be used in the fitting process. Should be NULL or a numeric vector. If non-NULL, weighted least squares is used for the level-specific regressions. |
| `kappa` | A vector of target effective degrees of freedom of the regression. |
| `q` | Thinning parameter representing the quantile cut-off (in terms of contributed variance) above which to consider levels in the hierarchy. This can used to reduce the number of levels in high-dimensional problems. Default is no thinning. |
| `intercept` | Should intercept be fitted. Default is `intercept=TRUE`. |

`standardize` Logical flag for `x` variable standardization prior to fitting the model. The coefficients are always returned on the original scale. Default is `standardize=TRUE`.

`nfolds` The number of folds for k-fold cross validation. Default is `nfolds=10`.

`foldid` An optional vector of values between `1` and `nfolds` identifying what fold each observation is in. If supplied, `nfolds` can be missing.

`partial_method` Indicate whether to use pairwise partial correlations, or shrinkage partial correlations.

`l2_penalty` Optional penalty for level-specific regressions (useful in high-dimensional case)

`...` Additional arguments passed to `hclust`.

## Details

This function fits an HFR to a grid of `kappa` hyperparameter values. The result is a matrix of coefficients with one column for each hyperparameter. By evaluating all hyperparameters in a single function, the speed of the cross-validation procedure is improved substantially (since level-specific regressions are estimated only once).

When `nfolds` > `1`, a cross validation is performed with shuffled data. Alternatively, test slices can be passed to the function using the `foldid` argument. The result of the cross validation is given by `best_kappa` in the output object.

## Value

A ’cv.hfr’ regression object.

## Author(s)

Johann Pfitzinger

## References

Pfitzinger, Johann (2024). Cluster Regularization via a Hierarchical Feature Regression. _Econometrics and Statistics_ (in press). URL https://doi.org/10.1016/j.ecosta.2024.01.003.

## See Also

`hfr`, `coef`, `plot` and `predict` methods

## Examples

```
x = matrix(rnorm(100 * 20), 100, 20)
y = rnorm(100)
fit = cv.hfr(x, y, kappa = seq(0, 1, by = 0.1))
coef(fit)
```

## hfr *Fit a hierarchical feature regression*

### Description

HFR is a regularized regression estimator that decomposes a least squares regression along a supervised hierarchical graph, and shrinks the edges of the estimated graph to regularize parameters. The algorithm leads to group shrinkage in the regression parameters and a reduction in the effective model degrees of freedom.

### Usage

```
hfr(
  x,
  y,
  weights = NULL,
  kappa = 1,
  q = NULL,
  intercept = TRUE,
  standardize = TRUE,
  partial_method = c("pairwise", "shrinkage"),
  l2_penalty = 0,
  ...
)
```

### Arguments

`x` Input matrix or data.frame, of dimension $(N \times p)$; each row is an observation vector.

`y` Response variable.

`weights` an optional vector of weights to be used in the fitting process. Should be NULL or a numeric vector. If non-NULL, weighted least squares is used for the level-specific regressions.

`kappa` The target effective degrees of freedom of the regression as a percentage of $p$.

`q` Thinning parameter representing the quantile cut-off (in terms of contributed variance) above which to consider levels in the hierarchy. This can used to reduce the number of levels in high-dimensional problems. Default is no thinning.

`intercept` Should intercept be fitted. Default is `intercept=TRUE`.

`standardize` Logical flag for x variable standardization prior to fitting the model. The coefficients are always returned on the original scale. Default is `standardize=TRUE`.

`partial_method` Indicate whether to use pairwise partial correlations, or shrinkage partial correlations.

`l2_penalty` Optional penalty for level-specific regressions (useful in high-dimensional case)

`...` Additional arguments passed to `hclust`.

## Details

Shrinkage can be imposed by targeting an explicit effective degrees of freedom. Setting the argument `kappa` to a value between `0` and `1` controls the effective degrees of freedom of the fitted object as a percentage of $p$. When `kappa` is `1` the result is equivalent to the result from an ordinary least squares regression (no shrinkage). Conversely, `kappa` set to `0` represents maximum shrinkage.

When $p > N$ `kappa` is a percentage of $(N - 2)$.

If no `kappa` is set, a linear regression with `kappa = 1` is estimated.

Hierarchical clustering is performed using `hclust`. The default is set to ward.D2 clustering but can be overridden by passing a method argument to `...`.

For high-dimensional problems, the hierarchy becomes very large. Setting `q` to a value below 1 reduces the number of levels used in the hierarchy. `q` represents a quantile-cutoff of the amount of variation contributed by the levels. The default (`q = NULL`) considers all levels.

When data exhibits multicollinearity it can be useful to include a penalty on the l2 norm in the level-specific regressions. This can be achieved by setting the `l2_penalty` parameter.

## Value

An 'hfr' regression object.

## Author(s)

Johann Pfitzinger

## References

Pfitzinger, Johann (2024). Cluster Regularization via a Hierarchical Feature Regression. _Econometrics and Statistics_ (in press). URL https://doi.org/10.1016/j.ecosta.2024.01.003.

## See Also

`cv.hfr`, `se.avg`, `coef`, `plot` and `predict` methods

## Examples

```
x = matrix(rnorm(100 * 20), 100, 20)
y = rnorm(100)
fit = hfr(x, y, kappa = 0.5)
coef(fit)
```

## plot.cv.hfr *Plot the dendrogram of an HFR model*

### Description

Plots the dendrogram of a fitted `cv.hfr` model. The heights of the levels in the dendrogram are given by a shrinkage vector, with a maximum (unregularized) overall graph height of $p$ (the number of covariates in the regression). Stronger shrinkage leads to a shallower hierarchy.

### Usage

```
## S3 method for class 'cv.hfr'
plot(x, kappa = NULL, show_details = TRUE, max_leaf_size = 3, ...)
```

### Arguments

| | |
|---|---|
| `x` | Fitted 'cv.hfr' model. |
| `kappa` | The hyperparameter used for plotting. If empty, the optimal value is used. |
| `show_details` | print model details on the plot. |
| `max_leaf_size` | maximum size of the leaf nodes. Default is `max_leaf_size=3`. |
| `...` | additional methods passed to `plot`. |

### Details

The dendrogram is generated using hierarchical clustering and modified so that the height differential between any two splits is the shrinkage weight of the lower split (ranging between `0` and `1`). With no shrinkage, all shrinkage weights are equal to `1` and the dendrogram has a height of $p$. With shrinkage the dendrogram has a height of $(\kappa \times p)$.

The leaf nodes are colored to indicate the coefficient sign, with the size indicating the absolute magnitude of the coefficients.

A color bar on the right indicates the relative contribution of each level to the coefficient of determination, with darker hues representing a larger contribution.

### Value

A plotted dendrogram.

### Author(s)

Johann Pfitzinger

### See Also

`cv.hfr`, `predict` and `coef` methods

## Examples

```
x = matrix(rnorm(100 * 20), 100, 20)
y = rnorm(100)
fit = cv.hfr(x, y, kappa = seq(0, 1, by = 0.1))
plot(fit, kappa = 0.5)
```

---

## plot.hfr *Plot the dendrogram of an HFR model*

---

### Description

Plots the dendrogram of a fitted hfr model. The heights of the levels in the dendrogram are given by a shrinkage vector, with a maximum (unregularized) overall graph height of $p$ (the number of covariates in the regression). Stronger shrinkage leads to a shallower hierarchy.

### Usage

```
## S3 method for class 'hfr'
plot(x, show_details = TRUE, confidence_level = 0, max_leaf_size = 3, ...)
```

### Arguments

x Fitted 'hfr' model.

show_details print model details on the plot.

confidence_level coefficients with a lower approximate statistical confidence are highlighted in the plot, see details. Default is confidence_level=0.

max_leaf_size maximum size of the leaf nodes. Default is max_leaf_size=3.

... additional methods passed to plot.

### Details

The dendrogram is generated using hierarchical clustering and modified so that the height differential between any two splits is the shrinkage weight of the lower split (ranging between 0 and 1). With no shrinkage, all shrinkage weights are equal to 1 and the dendrogram has a height of $p$. With shrinkage the dendrogram has a height of $(\kappa \times p)$.

The leaf nodes are colored to indicate the coefficient sign, with the size indicating the absolute magnitude of the coefficients.

The average standard errors along the branch of each coefficient can be used to highlight coefficients that are not statistically significant. When confidence_level > 0, branches with a lower confidence are plotted as dotted lines.

A color bar on the right indicates the relative contribution of each level to the coefficient of determination, with darker hues representing a larger contribution.

### Value

A plotted dendrogram.

### Author(s)

Johann Pfitzinger

### See Also

hfr, se.avg, predict and coef methods

### Examples

```
x = matrix(rnorm(100 * 20), 100, 20)
y = rnorm(100)
fit = hfr(x, y, kappa = 0.5)
plot(fit)
```

## predict.cv.hfr *Model predictions*

### Description

Predict values using a fitted cv.hfr model

### Usage

```
## S3 method for class 'cv.hfr'
predict(object, newdata = NULL, kappa = NULL, ...)
```

### Arguments

| | |
|---|---|
| object | Fitted 'cv.hfr' model. |
| newdata | Matrix or data.frame of new values for x at which predictions are to be made. |
| kappa | The hyperparameter used for prediction. If empty, the optimal value is used. |
| ... | additional methods passed to predict. |

### Details

Predictions are made by multiplying the newdata object with the estimated coefficients. The chosen hyperparameter value to use for predictions can be passed to the kappa argument.

### Value

A vector of predicted values.

### Author(s)

Johann Pfitzinger

### See Also

`hfr`, `cv.hfr` and `coef` methods

### Examples

```
x = matrix(rnorm(100 * 20), 100, 20)
y = rnorm(100)
fit = cv.hfr(x, y, kappa = seq(0, 1, by = 0.1))
predict(fit, kappa = 0.1)
```

## `predict.hfr` *Model predictions*

### Description

Predict values using a fitted `hfr` model

### Usage

```
## S3 method for class 'hfr'
predict(object, newdata = NULL, ...)
```

### Arguments

| | |
|---|---|
| `object` | Fitted 'hfr' model. |
| `newdata` | Matrix or data.frame of new values for `x` at which predictions are to be made. |
| `...` | additional methods passed to `predict`. |

### Details

Predictions are made by multiplying the `newdata` object with the estimated coefficients.

### Value

A vector of predicted values.

### Author(s)

Johann Pfitzinger

### See Also

`hfr`, `cv.hfr` and `coef` methods

## Examples

```
x = matrix(rnorm(100 * 20), 100, 20)
y = rnorm(100)
fit = hfr(x, y, kappa = 0.5)
predict(fit)
```

---

# print.cv.hfr *Print an HFR model*

---

## Description

Print summary statistics for a fitted cv.hfr model

## Usage

```
## S3 method for class 'cv.hfr'
print(x, ...)
```

## Arguments

| | |
|---|---|
| x | Fitted cv.hfr model. |
| ... | additional methods passed to print. |

## Details

The call that produced the object x is printed, following by a data.frame of summary statistics, including the effective degrees of freedom of the model, the R.squared and the regularization parameter.

## Value

Summary statistics of HFR model

## Author(s)

Johann Pfitzinger

## See Also

hfr, cv.hfr and coef methods

## Examples

```
x = matrix(rnorm(100 * 20), 100, 20)
y = rnorm(100)
fit = cv.hfr(x, y, kappa = seq(0, 1, by = 0.1))
print(fit)
```

## print.hfr *Print an HFR model*

### Description

Print summary statistics for a fitted `hfr` model

### Usage

```
## S3 method for class 'hfr'
print(x, ...)
```

### Arguments

| | |
|---|---|
| `x` | Fitted `hfr` model. |
| `...` | additional methods passed to `print`. |

### Details

The call that produced the object `x` is printed, following by a `data.frame` of summary statistics, including the effective degrees of freedom of the model, the R.squared and the regularization parameter.

### Value

Summary statistics of HFR model

### Author(s)

Johann Pfitzinger

### See Also

`hfr`, `cv.hfr` and `coef` methods

### Examples

```
x = matrix(rnorm(100 * 20), 100, 20)
y = rnorm(100)
fit = hfr(x, y, kappa = 0.5)
print(fit)
```

## se.avg *Calculate approximate standard errors for a fitted HFR model*

### Description

This function computes the weighted average standard errors across levels using Burnham & Anderson (2004).

### Usage

```
se.avg(object)
```

### Arguments

| | |
|---|---|
| `object` | Fitted `hfr` model. |

### Details

The HFR computes linear regressions over several levels of an estimated hierarchy. By averaging the standard errors across hierarchical levels, an indication can be obtained about the average significance of the variables.

Standard errors are understated, since the uncertainty in the hierarchy estimation is not reflected.

### Value

A vector of standard errors.

### Author(s)

Johann Pfitzinger

### References

Pfitzinger, J. (2022). Cluster Regularization via a Hierarchical Feature Regression. arXiv 2107.04831[statML]

Burnham, K. P. and Anderson, D. R. (2004). Multimodel inference - understanding AIC and BIC in model selection. Sociological Methods & Research 33(2): 261-304.

### See Also

`hfr` method

### Examples

```
x = matrix(rnorm(100 * 20), 100, 20)
y = rnorm(100)
fit = hfr(x, y, kappa = 0.5)
se.avg(fit)
```

# Index