Package ‘abundant’

October 12, 2022

Type Package
Title High-Dimensional Principal Fitted Components and Abundant Regression
Version 1.2
Date 2022-01-04
Author Adam J. Rothman
Maintainer Adam J. Rothman <arothman@umn.edu>
Depends R (>= 2.10), glasso
Description Fit and predict with the high-dimensional principal fitted components model. This model is described by Cook, Forzani, and Rothman (2012) <doi:10.1214/11-AOS962>.
License GPL-2
NeedsCompilation yes
Repository CRAN
Date/Publication 2022-01-04 15:30:19 UTC

R topics documented:

abundant-package .................................................. 1
fit.pfc ................................................................. 2
pred.response ....................................................... 5

Index 7

abundant-package Abundant regression and high-dimensional principal fitted components

Description

Fit and predict with the high-dimensional principal fitted components model.
Details

The main functions are `fit.pfc, pred.response`.

Author(s)

Adam J. Rothman

Maintainer: Adam J. Rothman <arothman@umn.edu>

References


---

**fit.pfc**

*Fit a high-dimensional principal fitted components model using the method of Cook, Forzani, and Rothman (2012).*

---

Description

Let \((x_1, y_1), \ldots, (x_n, y_n)\) denote the \(n\) measurements of the predictor and response, where \(x_i \in \mathbb{R}^p\) and \(y_i \in \mathbb{R}\). The model assumes that these measurements are a realization of \(n\) independent copies of the random vector \((X, Y)'\), where

\[
X = \mu_X + \Gamma \beta (f(Y) - \mu_f) + \epsilon,
\]

\(\mu_X \in \mathbb{R}^p; \Gamma \in \mathbb{R}^{p \times d}\) with rank \(d; \beta \in \mathbb{R}^{d \times r}\) with rank \(d; f : \mathbb{R} \rightarrow \mathbb{R}^r\) is a known vector valued function; \(\mu_f = \mathbb{E}\{f(Y)\}; \epsilon \sim N_p(0, \Delta)\); and \(Y\) is independent of \(\epsilon\). The central subspace is \(\Delta^{-1}\text{span}(\Gamma)\).

This function computes estimates of these model parameters by imposing constraints for identifiability. The mean parameters \(\mu_X\) and \(\mu_f\) are estimated with \(\bar{x} = n^{-1} \sum_{i=1}^n x_i\) and \(\bar{f} = n^{-1} \sum_{i=1}^n f(y_i)\). Let \(\hat{\Phi} = n^{-1} \sum_{i=1}^n (f(y_i) - \bar{f})(f(y_i) - \bar{f})'\), which we require to be positive definite. Given a user-specified weight matrix \(\hat{W}\), let

\[
(\hat{\Gamma}, \hat{\beta}) = \arg \min_{G \in \mathbb{R}^{p \times d}, B \in \mathbb{R}^{d \times r}} \sum_{i=1}^n [x_i - \bar{x} - GB\{f(y_i) - \bar{f}\}]' \hat{W} [x_i - \bar{x} - GB\{f(y_i) - \bar{f}\}],
\]

subject to the constraints that \(G' \hat{W} G\) is diagonal and \(B\hat{\Phi}B' = I\). The sufficient reduction estimate \(\hat{R} : \mathbb{R}^p \rightarrow \mathbb{R}^d\) is defined by

\[
\hat{R}(x) = (\hat{\Gamma}' \hat{W} \hat{\Gamma})^{-1} \hat{\Gamma}' \hat{W}(x - \bar{x}).
\]

Usage

```r
fit.pfc(X, y, r=4, d=NULL, F.user=NULL, weight.type=c("sample", "diag", "L1"), lam.vec=NULL, kfold=5, silent=TRUE, qrtol=1e-10, cov.tol=1e-4, cov.maxit=1e3, NPERM=1e3, level=0.01)
```
Arguments

\(X\) The predictor matrix with \(n\) rows and \(p\) columns. The \(i\)th row is \(x_i\) defined above.

\(y\) The vector of measured responses with \(n\) entries. The \(i\)th entry is \(y_i\) defined above.

\(r\) When polynomial basis functions are used (which is the case when \(\text{F.user=NULL}\)), \(r\) is the polynomial order, i.e., \(f(y) = (y, y^2, \ldots, y^r)'\). The default is \(r=4\). This argument is not used when \(\text{F.user}\) is specified.

\(d\) The dimension of the central subspace defined above. This must be specified by the user when \(\text{weight.type="L1"}\). If unspecified by the user this function will use the sequential permutation testing procedure, described in Section 8.2 of Cook, Forzani, and Rothman (2012), to select \(d\).

\(\text{F.user}\) A matrix with \(n\) rows and \(r\) columns, where the \(i\)th row is \(f(y_i)\) defined above. This argument is optional, and will typically be used when polynomial basis functions are not desired.

\(\text{weight.type}\) The type of weight matrix estimate \(\hat{W}\) to use. Let \(\hat{\Delta}\) be the observed residual sample covariance matrix for the multivariate regression of \(X\) on \(f(Y)\) with \(n-r-1\) scaling. There are three options for \(\hat{W}\):

- \(\text{weight.type="sample"}\) uses a Moore-Penrose generalized inverse of \(\hat{\Delta}\) for \(\hat{W}\); when \(p \leq n-r-1\) this becomes the inverse of \(\hat{\Delta}\);
- \(\text{weight.type="diag"}\) uses the inverse of the diagonal matrix with the same diagonal as \(\hat{\Delta}\) for \(\hat{W}\);
- \(\text{weight.type="L1"}\) uses the L1-penalized inverse of \(\hat{\Delta}\) described in equation (5.4) of Cook, Forzani, and Rothman (2012). In this case, \(\text{lam.vec}\) and \(d\) must be specified by the user. The glasso algorithm of Friedman et al. (2008) is used through the R package glasso.

\(\text{lam.vec}\) A vector of candidate tuning parameter values to use when \(\text{weight.type="L1"}\). If this vector has more than one entry, then \(k\text{fold}\) cross validation will be performed to select the optimal tuning parameter value.

\(k\text{fold}\) The number of folds to use in cross-validation to select the optimal tuning parameter when \(\text{weight.type="L1"}\). Only used if \(\text{lam.vec}\) has more than one entry.

\(\text{silent}\) Logical. When \(\text{silent=FALSE}\), progress updates are printed.

\(\text{qr.tol}\) The tolerance for calls to \(\text{qr.solve()}\).

\(\text{cov.tol}\) The convergence tolerance for the QUIC algorithm used when \(\text{weight.type="L1"}\).

\(\text{cov.maxit}\) The maximum number of iterations allowed for the QUIC algorithm used when \(\text{weight.type="L1"}\).

\(\text{NPERM}\) The number of permutations to used in the sequential permutation testing procedure to select \(d\). Only used when \(d\) is unspecified.

\(\text{level}\) The significance level to use to terminate the sequential permutation testing procedure to select \(d\).

Details

Value

A list with

- `Gamhat` this is $\hat{\Gamma}$ described above.
- `bhat` this is $\hat{\beta}$ described above.
- `Rmat` this is $\hat{W}(\hat{\Gamma}'\hat{W}\hat{\Gamma})^{-1}$.
- `What` this is $\hat{W}$ described above.
- `d` this is $d$ described above.
- `r` this is $r$ described above.
- `GWG` this is $\hat{\Gamma}'\hat{W}\hat{\Gamma}$
- `fc` a matrix with $n$ rows and $r$ columns where the $i$th row is $f(y_i) - \bar{f}$.
- `Xc` a matrix with $n$ rows and $p$ columns where the $i$th row is $x_i - \bar{x}$.
- `y` the vector of $n$ response measurements.
- `mx` this is $\bar{x}$ described above.
- `mf` this is $\bar{f}$ described above.
- `best.lam` this is selected tuning parameter value used when `weight.type="L1"`, will be `NULL` otherwise.
- `lam.vec` this is the vector of candidate tuning parameter values used when `weight.type="L1"`, will be `NULL` otherwise.
- `err.vec` this is the vector of validation errors from cross validation, one error for each entry in `lam.vec`. Will be `NULL` unless `weight.type="L1"` and `lam.vec` has more than one entry.
- `test.info` a dataframe that summarizes the results from the sequential testing procedure. Will be `NULL` unless `d` is unspecified.

Author(s)

Adam J. Rothman

References


See Also

`pred.response`
Examples

```r
set.seed(1)
n = 20
p = 30
d = 2
y = sqrt(12) * runif(n)
Gam = matrix(rnorm(p * d), nrow=p, ncol=d)
beta = diag(2)
E = matrix(0.5 * rnorm(n * p), nrow=n, ncol=p)
V = matrix(c(1, sqrt(12), sqrt(12), 12.8, nrow=2, ncol=2)
tmp = eigen(V, symmetric=TRUE)
V.msqrt = tcrossprod(tmp$vec * rep(tmp$val^(-0.5), each=2), tmp$vec)
Fyc = cbind(y - sqrt(3), y^2 - 4) %*% V.msqrt
X = X + Fyc %*% t(beta) %*% t(Gam) + E
fit = fit.pfc(X = X, y = y, r = 3, weight.type = "sample")
## display hypothesis testing information for selecting d
fit$test.info
## make a response versus fitted values plot
plot(pred.response(fit), y)
```

---

**pred.response**

Predict the response with the fitted high-dimensional principal fitted components model

**Description**

Let \( x \in \mathbb{R}^p \) denote the values of the \( p \) predictors. This function computes \( \hat{E}(Y|X = x) \) using equation (8.1) of Cook, Forzani, and Rothman (2012).

**Usage**

`pred.response(fit, newx=NULL)`

**Arguments**

- `fit` The object returned by `fit.pfc()`.
- `newx` A matrix with \( N \) rows and \( p \) columns where each row is an instance of \( x \) described above. If this argument is unspecified, then the fitted values are returned, i.e., \( newx=X \), where \( X \) was the predictor matrix used in the call to `fit.pfc()`.

**Details**

See Cook, Forzani, and Rothman (2012) for more information.

**Value**

A vector of response prediction with `nrow(newx)` entries.
Author(s)

Adam J. Rothman

References


See Also

fit.pfc

Examples

set.seed(1)
n=25
p=50
d=1
t = matrix(rnorm(p*d), nrow=p, ncol=d)
y=rnorm(n)
y = y
E=matrix(rnorm(n*p), nrow=n, ncol=p)
X=fy%*%t(true.G) + E
fit=fit.pfc(X=X, r=4, d=d, y=y, weight.type="diag")
fitted.values=pred.response(fit)
mean((y-fitted.values)^2)
plot(fitted.values, y)

n.new=100
y.new=rnorm(n.new)
y.new=y.new
E.new=matrix(rnorm(n.new*p), nrow=n.new, ncol=p)
X.new = fy.new%*%t(true.G) + E.new
mean((y.new - pred.response(fit, newx=X.new))^2)
Index

abundant (abundant-package), 1
abundant-package, 1
fit.pfc, 2, 2, 6
pred.response, 2, 4, 5