Package ‘shrink’

October 31, 2023

Title  Global, Parameterwise and Joint Shrinkage Factor Estimation
Version  1.2.3
Description  The predictive value of a statistical model can often be improved by applying shrinkage methods. This can be achieved, e.g., by regularized regression or empirical Bayes approaches. Various types of shrinkage factors can also be estimated after a maximum likelihood. While global shrinkage modifies all regression coefficients by the same factor, parameterwise shrinkage factors differ between regression coefficients. With variables which are either highly correlated or associated with regard to contents, such as several columns of a design matrix describing a nonlinear effect, parameterwise shrinkage factors are not interpretable and a compromise between global and parameterwise shrinkage, termed 'joint shrinkage', is a useful extension. A computational shortcut to resampling-based shrinkage factor estimation based on DFBETA residuals can be applied. Global, parameterwise and joint shrinkage for models fitted by lm(), glm(), coxph(), or mfp() is available.

Depends  R (>= 3.2.2)
License  GPL-3
Imports  survival, MASS, rms, mfp
LazyData  true
Suggests  aod, knitr, rmarkdown
VignetteBuilder  knitr
RoxygenNote  7.2.3
Encoding  UTF-8
URL  https://github.com/biometrician/shrink
BugReports  https://github.com/biometrician/shrink/issues
NeedsCompilation  no
Author  Daniela Dunkler [aut, cre],
        Georg Heinze [aut]
Maintainer  Daniela Dunkler <daniela.dunkler@meduniwien.ac.at>
Repository  CRAN
Date/Publication  2023-10-31 12:30:02 UTC
Description

The predictive value of a statistical model can often be improved by applying shrinkage methods. This can be achieved, e.g., by regularized regression or empirical Bayes approaches. Various types of shrinkage factors can also be estimated after a maximum likelihood. While global shrinkage modifies all regression coefficients by the same factor, parameterwise shrinkage factors differ between regression coefficients. With variables which are either highly correlated or associated with regard to contents, such as several columns of a design matrix describing a nonlinear effect or two main effects and their pairwise interaction term, parameterwise shrinkage factors are not interpretable and a compromise between global and parameterwise shrinkage, termed 'joint shrinkage', is a useful extension. A computational shortcut to resampling-based shrinkage factor estimation based on DFBETA residuals can be applied. Global, parameterwise and joint shrinkage for models fitted by \texttt{lm}, \texttt{glm}, \texttt{coxph}, and \texttt{mfp} is available.

Details

Functions included in the \texttt{shrink}-package:

\begin{verbatim}
shrink             a function to compute global, parameterwise and joint post-estimation shrinkage factors of fit objects of class \texttt{lm}, \texttt{glm}, \texttt{coxph}, or \texttt{mfp}.
coef.shrink        returns shrunken regression coefficients from objects of class \texttt{shrink}.
predict.shrink     obtains predictions from shrunken regression coefficients from objects of class \texttt{shrink}.
vcov.shrink        returns the variance-covariance matrix of shrinkage factors.
print.shrink       prints objects of class \texttt{shrink}.
summary.shrink     summary of objects of class \texttt{shrink}.
\end{verbatim}
Sauerbrei (1999) suggested that before estimating parameterwise shrinkage factors, the data should be standardized to have a mean of 0 and unit variance.

References


See Also

`shrink`, `coef.shrink`, `predict.shrink`, `print.shrink`, `summary.shrink`, `vcov.shrink`, `deepvein`

Examples

```r
# with glm, family = binomial
set.seed(888)
intercept <- 1
beta <- c(0.5, 1.2)
n <- 200
x1 <- rnorm(n, mean = 1, sd = 1)
x2 <- rbinom(n, size = 1, prob = 0.3)
prob <- exp(linpred) / (1 + exp(linpred))
runis <- runif(n, min = 0, max = 1)
ytest <- ifelse(runis < prob, yes = 1, no = 0)
simdat <- data.frame(cbind(y = ifelse(runis < prob, 1, 0), x1, x2))
fit <- glm(y ~ x1 + x2, family = binomial, data = simdat, x = TRUE)
summary(fit)

global <- shrink(fit, type = "global", method = "dfbeta")
print(global)
coef(global)

shrink(fit, type = "parameterwise", method = "dfbeta")

shrink(fit, type = "parameterwise", method = "dfbeta", join = list(c("x1", "x2")))

#shrink(fit, type = "global", method = "jackknife")
#shrink(fit, type = "parameterwise", method = "jackknife")
#shrink(fit, type = "parameterwise", method = "jackknife",
#      join = list(c("x1", "x2")))
```
# For more examples see shrink

## S3 method for class 'shrink'

```
coef(object, ...)  
```

### Arguments

- `object` an object of class `shrink`.
- `...` further arguments.

### Value

A vector with shrunken regression coefficients

### See Also

`shrink`, `print.shrink`, `predict.shrink`, `summary.shrink`, `vcov.shrink`

---

**deepvein**

### Description

A data frame containing time to recurrence of thrombosis and several potential prognostic factors measured at baseline for 929 individuals with deep vein thrombosis or unprovoked pulmonary embolism. 147 events of recurrence were observed during a median follow-up time of 37.8 months.
Format

The data frame contains observations of 929 individuals and the following variables:

- **pnr**: patient number.
- **time**: time to recurrence of thrombosis or end of study in months.
- **status**: = 1 recurrence of thrombosis.
- **sex**: gender.
- **fimut**: factor II G20210A mutation.
- **fvleid**: factor V Leiden mutation.
- **log2ddim**: log2-transformed D-dimer.
- **bmi**: body mass index.
- **durther**: duration of anticoagulation therapy.
- **age**: age in years.
- **loc**: location of first thrombosis: pulmonary embolism (PE), distal, or proximal deep vein thrombosis.

Note

The data are a modified and partly simulated version of the data set used by Eichinger et al. (2010) and are available under a GPL-2 license.

References


Examples

data("deepvein")
summary(deepvein)
Description
A data frame containing the observations from the GBSG study.

Usage
GBSG

Format
This data frame contains the observations of 686 women:

- **id**: patient id.
- **htreat**: hormonal therapy, a factor at two levels 0 (no) and 1 (yes).
- **age**: of the patients in years.
- **menostat**: menopausal status, a factor at two levels 1 (premenopausal) and 2 (postmenopausal).
- **tumsize**: tumor size (in mm).
- **tumgrad**: tumor grade, an ordered factor at levels 1 < 2 < 3.
- **posnodal**: number of positive nodes.
- **prm**: progesterone receptor (in fmol).
- **esm**: estrogen receptor (in fmol).
- **rfst**: recurrence free survival time (in days).
- **cens**: censoring indicator (0 censored, 1 event).

References


Examples
```r
data("GBSG")
summary(GBSG)
```
predict.shrink

Predict Method for Objects of Class shrink

Description

Obtains predictions from shrunken regression coefficients from an object of class shrink. This class of objects is returned by the shrink function. Objects of this class have methods for the functions coef, predict, print, summary, and vcov.

Usage

```r
## S3 method for class 'shrink'
predict(
  object,
  newdata = NULL,
  type = c("link", "response", "lp", "risk", "expected", "terms"),
  shrinktype = NULL,
  terms = NULL,
  na.action = na.pass,
  collapse,
  safe = FALSE,
  ...
)
```

Arguments

- `object` an object of class shrink.
- `newdata` a data frame for which predictions are obtained, otherwise predictions are based on the data stored in object.
- `type` the type of prediction required.
- `shrinktype` the type of shrinkage requested, if the object was obtained with `type = "all"`, either "parameterwise" or "global".
- `terms` with `type = "terms"` by default all terms are returned. A character vector specifies which terms are to be returned.
- `na.action` function determining what should be done with missing values in `newdata`. The default is to include all observations.
- `collapse` if `family = coxph` or `Cox`, an optional vector of subject identifiers. If specified, the output will contain one entry per subject rather than one entry per observation.
- `safe` option from predict.mfp.
- `...` additional arguments to be passed to methods.

Value

A vector or matrix of predictions.
Note

If object was obtained using type = "all", shrinktype specifies for which type of shrinkage predictions are requested. shrinktype will be ignored if object was obtained using either type = "parameterwise" or type = "global".

See Also

shrink, coef.shrink, print.shrink, summary.shrink, vcov.shrink

Examples

data("GBSG")
library("mfp")

fit <- mfp(Surv(rfst, cens) ~ fp(age, df = 4, select = 0.05) +
   fp(prm, df = 4, select = 0.05), family = cox, data = GBSG)

dfbeta.global <- shrink(fit, type = "global", method = "dfbeta")
dfbeta.pw <- shrink(fit, type = "parameterwise", method = "dfbeta")
dfbeta.join <- shrink(fit, type = "parameterwise", method = "dfbeta",
   join=list(c("age.1", "age.2")))

age <- 30:80
newdat <- data.frame(age = age, prm = 0)
refdat <- data.frame(age = 50, prm = 0)

# unshrunken
plot(age, predict(fit, newdata = newdat, type = "lp") -
     predict(fit, newdata = refdat, type = "lp"), xlab = "Age",
     ylab = "Log hazard relative to 50 years", type = "l", lwd = 2)

# globally shrunken
lines(age, predict(dfbeta.global, newdata = newdat, type = "lp") -
      predict(dfbeta.global, newdata = refdat, type = "lp"), lty = 3, col = "red", lwd = 2)

# jointly shrunken
lines(age, predict(dfbeta.join, newdata = newdat, type = "lp") -
      predict(dfbeta.join, newdata = refdat, type = "lp"), lty = 4, col = "blue", lwd = 2)

# parameterwise shrunken
lines(age, predict(dfbeta.pw, newdata = newdat, type = "lp") -
      predict(dfbeta.pw, newdata = refdat, type = "lp"), lty = 2, col = "green", lwd = 2)

legend("topright", lty = c(1, 3, 4, 2), title = "SHRINKAGE",
      legend = c("No", "Global", "Joint", "Parameterwise"), inset = 0.01, bty = "n",
      col = c("black", "red", "blue", "green"), lwd = 2)
print.shrink

Print Method for Objects of Class shrink

Description

This class of objects is returned by the shrink function. Objects of this class have methods for the functions coef, predict, print, summary, and vcov.

Usage

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

Arguments

x

object of class shrink.

...

further arguments.

See Also

shrink, coef.shrink, predict.shrink, summary.shrink, vcov.shrink

shrink

Global, Parameterwise and Joint Shrinkage of Regression Coefficients

Description

Obtain global, parameterwise and joint post-estimation shrinkage factors for regression coefficients from fit objects of class lm, glm, coxph, or mfp.

Usage

shrink(
    fit,
    type = c("parameterwise", "global", "all"),
    method = c("jackknife", "dfbeta"),
    join = NULL,
    notes = TRUE,
    postfit = TRUE
)

Arguments

**fit**
A fit object of class `lm`, `glm`, `coxph`, or `mfp`. The fit object must have been called with `x = TRUE` (and `y = TRUE` in case of class `lm`).

**type**
The type of shrinkage, either "parameterwise" (default), "global" shrinkage, or "all".

**method**
The method of shrinkage estimation, either "jackknife" (based on leave-one-out resampling, default) or "dfbeta" (excellent approximation based on DFBETA residuals).

**join**
Compute optional joint shrinkage factors for sets of specified columns of the design matrix, if `type = "parameterwise"`. See details.

**notes**
Print notes. Default is TRUE.

**postfit**
Obtain fit with shrunken regression coefficients. This option is only available for models without an intercept. Default is TRUE.

Details

While global shrinkage modifies all regression coefficients by the same factor, parameterwise shrinkage factors differ between regression coefficients. With variables which are either highly correlated or associated with regard to contents, such as several columns of a design matrix describing a nonlinear effect, parameterwise shrinkage factors are not interpretable. Joint shrinkage of a set of such associated design variables will give one common shrinkage factor for this set.

Joint shrinkage factors may be useful when analysing highly correlated and/or such associated columns of the design matrix, e.g. dummy variables corresponding to a categorical explanatory variable with more than two levels, two variables and their pairwise interaction term, or several transformations of an explanatory variable enabling estimation of nonlinear effects. The analyst can define 'joint' shrinkage factors by specifying the `join` option if `type = "parameterwise"`. `join` expects a list with at least one character vector including the names of the columns of the design matrix for which a joint shrinkage factor is requested. For example the following specification of `join = list(c("dummy1", "dummy2", "dummy3"), c("main1", "main2", "interaction"), c("varX.fp1", "varX.fp2"))` requests the joint shrinkage factors for a) "dummy1", "dummy2" and "dummy3", b) "main1", "main2" and "interaction" and c) "varX.fp1" and "varX.fp2".

**Restricted cubic splines using rcs:** shrink also works for models incorporating restricted cubic splines computed with the `rcs` function from the `rms` package. A joint shrinkage factor of explanatory variable `varX` transformed with `rcs` can be obtained by `join = list(c("rcs(varX)"))` or by stating the names of the `rcs`-transformed variables as given in the respective fit object. (These two notations should not be mixed within one call to `shrink`.)

**Jackknife versus DFBETA method:** For linear regression models (`lm` or `glm` with `family = "gaussian"`) shrinkage factors obtained by Jackknife and the DFBETA approximation will be identical. For all other types of regression, the computational effort of estimating shrinkage factors may be greatly reduced by using `method = "dfbeta"` instead. However, for (very) small data sets `method = "jackknife"` may be of advantage, as the use of DFBETA residuals may underestimate the influence of some highly influential observations.

**Shrunken intercept:** A shrunken intercept is estimated as follows: For all columns of the design matrix except for the intercept the shrinkage factors are multiplied with the respective regression coefficients and a linear predictor is computed. Then the shrunken intercept is estimated by modeling `fit$y ~ offset(linear predictor)`. 
For regression models without an intercept, i.e., fit objects of class coxph, the shrunken regression coefficients can be directly estimated. This postfit is retained in the $postfit component of the shrink object.

**Value**

shrink returns an object with the following components:

- **ShrinkageFactors**: a vector of shrinkage factors of regression coefficients.
- **ShrinkageFactorsVCOV**: the covariance matrix of the shrinkage factors.
- **ShrunkenRegCoeff**: a vector with the shrunken regression coefficients.
- **postfit**: an optional postfit model with shrunken regression coefficients and associated standard errors for models without an intercept.
- **fit**: the original (unshrunken) fit object.
- **type**: the requested shrinkage type.
- **method**: the requested shrinkage method.
- **join**: the requested joint shrinkage factors.
- **call**: the function call.

If type = "all" then the object returned by shrink additionally contains

- **global**: a list with the following elements: ShrinkageFactors, ShrinkageFactorsVCOV and ShrunkenRegCoeff.
- **parameterwise**: a list with the following elements: ShrinkageFactors, ShrinkageFactorsVCOV and ShrunkenRegCoeff.
- **joint**: an optional list with the following elements: ShrinkageFactors, ShrinkageFactorsVCOV and ShrunkenRegCoeff.

**Note**

For fit objects of class mfp with family != cox regression coefficients of fit (obtained by coef(fit)) and fit$fit may not always be identical, because of mfp’s pretransformation applied to the explanatory variables in the model. The shrink function uses a) the names as given in names(coef(fit)) and b) the regression coefficients as given in summary(fit) which correspond to the pretransformed explanatory variables.

**References**


**See Also**

coeff.shrink, predict.shrink, print.shrink, summary.shrink, vcov.shrink

**Examples**

```r
## Example with mfp (family = cox)
```
data("GBSG")
library("mfp")
fit1 <- mfp(Surv(rfst, cens) ~ fp(age, df = 4, select = 0.05) +
fp(prm, df = 4, select = 0.05), family = cox, data = GBSG)
shrink(fit1, type = "global", method = "dfbeta")
dfbeta.pw <- shrink(fit1, type = "parameterwise", method = "dfbeta")
dfbeta.pw

dfbeta.pw$postfit

# correlations between shrinkage factors and standard errors of shrinkage factors
 cov2cor(dfbeta.pw$ShrinkageFactorsVCOV)
sqrt(diag(dfbeta.pw$ShrinkageFactorsVCOV))

shrink(fit1, type = "parameterwise", method = "dfbeta",
  join = list(c("age.1", "age.2")))

#shrink(fit1, type = "global", method = "jackknife")
#shrink(fit1, type = "parameterwise", method = "jackknife")
#shrink(fit1, type = "parameterwise", method = "jackknife",
#  join = list(c("age.1", "age.2")))

# obtain global, parameterwise and joint shrinkage with one call to 'shrink'
shrink(fit1, type = "all", method = "dfbeta",
  join = list(c("age.1", "age.2")))

## Example with rcs
library("rms")
fit2 <- coxph(Surv(rfst, cens) ~ rcs(age) + log(prm + 1), data = GBSG, x = TRUE)
shrink(fit2, type = "global", method = "dfbeta")
shrink(fit2, type = "parameterwise", method = "dfbeta")
shrink(fit2, type = "parameterwise", method = "dfbeta",
  join = list(c("rcs(age)")))
shrink(fit2, type = "parameterwise", method = "dfbeta",
  join = list(c("rcs(age)"), c("log(prm + 1)"))

## Examples with glm & mfp (family = binomial)
set.seed(888)
intercept <- 1
beta <- c(0.5, 1.2)
n <- 1000
x1 <- rnorm(n, mean = 1, sd = 1)
x2 <- rbinom(n, size = 1, prob = 0.3)
prob <- exp(linpred) / (1 + exp(linpred))
runis <- runif(n, 0, 1)
ytest <- ifelse(runis < prob, yes = 1, no = 0)
simdat <- data.frame(cbind(y = ifelse(runis < prob, 1, 0), x1, x2))
fit3 <- glm(y ~ x1 + x2, family = binomial, data = simdat, x = TRUE)
summary(fit3)

shrink(fit3, type = "global", method = "dfbeta")
shrink(fit3, type = "parameterwise", method = "dfbeta")
shrink(fit3, type = "parameterwise", method = "dfbeta", join = list(c("x1", "x2")))

utils::data("Pima.te", package="MASS")
utils::data("Pima.tr", package="MASS")
Pima <- rbind(Pima.te, Pima.tr)
fit4 <- mfp(type ~ npreg + glu + bmi + ped + fp(age, select = 0.05),
            family = binomial, data = Pima)
summary(fit4)

shrink(fit4, type = "global", method = "dfbeta")
shrink(fit4, type = "parameterwise", method = "dfbeta")

# fit objects of class mfp: for 'join' use variable names as given in 'names(coef(fit4))'
shrink(fit4, type = "parameterwise", method = "dfbeta", join = list(c("age.1")))

## Examples with glm & mfp (family = gaussian) and lm
utils::data("anorexia", package = "MASS")
contrasts(anorexia$Treat) <- contr.treatment(n = 3, base = 2)
fit5 <- glm(Postwt ~ Prewt + Treat, family = gaussian, data = anorexia, x = TRUE)

shrink(fit5, type = "global", method = "dfbeta")
# which is identical to the more time-consuming jackknife approach:
# shrink(fit5, type = "global", method = "jackknife")

shrink(fit5, type = "parameterwise", method = "dfbeta")
shrink(fit5, type = "parameterwise", method = "dfbeta",
       join = list(c("Treat1", "Treat3")))

fit6 <- lm(Postwt ~ Prewt + Treat, data = anorexia, x = TRUE, y = TRUE)

shrink(fit6, type = "global", method = "dfbeta")
shrink(fit6, type = "parameterwise", method = "dfbeta")
shrink(fit6, type = "parameterwise", method = "dfbeta",
       join = list(c("Treat1", "Treat3")))

utils::data("GAGurine", package = "MASS")
fit7 <- mfp(Age ~ fp(GAG, select = 0.05), family = gaussian, data = GAGurine)
summary(fit7)

shrink(fit7, type = "global", method = "dfbeta")
shrink(fit7, type = "parameterwise", method = "dfbeta")
# fit objects of class mfp: for 'join' use variable names as given in 'names(coef(fit7))'
shrink(fit7, type = "parameterwise", method = "dfbeta",
       join = list(c("GAG.1", "GAG.2")))
Summary Method for Objects of Class shrink

Description

This class of objects is returned by the shrink function. Objects of this class have methods for the functions coef, predict, print, summary, and vcov.

Usage

## S3 method for class 'shrink'
summary(object, digits = 6, ...)

Arguments

object an object of class shrink.
digits integer, used for number formatting with signif().
... further arguments.

Value

A matrix with regression coefficients of the original fit, corresponding shrinkage factors and shrunken regression coefficients.

See Also

shrink, coef.shrink, print.shrink, predict.shrink, vcov.shrink

Calculate Variance-Covariance Matrix of Shrinkage Factors for Objects of Class shrink

Description

This class of objects is returned by the shrink function. Objects of this class have methods for the functions coef, predict, print, summary, and vcov.

Usage

## S3 method for class 'shrink'
vcov(object, digits = 6, ...)
vcov.shrink

Arguments

  object    object of class shrink.
  digits    integer, used for number formatting with `signif()`.
  ...      further arguments.

Value

  A matrix of the estimated covariances between the obtained shrinkage factors.

See Also

  `shrink`, `coef.shrink`, `predict.shrink`, `print.shrink`, `summary.shrink`
Index

* **datasets**
  deepvein, 4
  GBSG, 6
  coef.shrink, 3, 4, 8, 9, 11, 14, 15
  deepvein, 3, 4
  GBSG, 6
  predict.shrink, 3, 4, 7, 9, 11, 14, 15
  print.shrink, 3, 4, 8, 9, 11, 14, 15
  rcs, 10
  shrink, 3, 4, 8, 9, 9, 14, 15
  shrink-package, 2
  signif, 14, 15
  summary.shrink, 3, 4, 8, 9, 11, 14, 15
  vcov.shrink, 3, 4, 8, 9, 11, 14, 14