

# Package ‘shrink’

July 2, 2014

**Type** Package

**Title** Global, Parameterwise, and Joint Post-Estimation Shrinkage

**Version** 1.1

**Date** 2013-10-21

**Author** Daniela Dunkler, Georg Heinze

**Maintainer** Daniela Dunkler <daniela.dunkler@meduniwien.ac.at>

**Description** Post-estimation shrinkage of regression coefficients in statistical modeling can be used to correct for the overestimation of regression coefficients caused by variable selection. While global shrinkage modifies all regression coefficients by the same factor, parameterwise shrinkage factors differ between regression coefficients. With highly correlated or semantically related variables, 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 is applied. Global, parameterwise, and joint shrinkage for models fitted by lm, glm, coxph, or mfp is available.

**Suggests** survival, mfp, rms, MASS

**License** GPL-2

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2013-10-21 19:46:47

## R topics documented:

shrink-package . . . . .	2
coef.shrink . . . . .	4
deepvein . . . . .	4

predict.shrink . . . . .	6
print.shrink . . . . .	8
shrink . . . . .	8

<b>Index</b>	<b>13</b>
--------------	-----------

---

shrink-package	<i>Global, Parameterwise, and Joint Post-Estimation Shrinkage</i>
----------------	---

---

## Description

In statistical modeling post-estimation shrinkage can be used to correct for the overestimation of regression coefficients caused by variable selection. While global shrinkage modifies all regression coefficients by the same factor, parameterwise shrinkage factors differ between regression coefficients. With highly correlated or semantically related variables, 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 is applied. Global, parameterwise, and joint shrinkage for models fitted by `lm`, `glm`, `coxph`, and `mfp` is available.

## Details

Package: shrink  
 Type: Package  
 Version: 1.1.  
 Date: 2013-10-21  
 License: GPL-2

Functions contained in the shrink-package are

<code>shrink</code>	a function to compute global, parameterwise, and joint post-estimation shrinkage factors of fit objects of class <code>lm</code> , <code>glm</code> , <code>coxph</code> , or <code>mfp</code> .
<code>coef.shrink</code>	returns shrunken regression coefficients from objects of class <code>shrink</code> .
<code>predict.shrink</code>	obtains predictions from shrunken regression coefficients from a fit object of class <code>shrink</code> .
<code>print.shrink</code>	prints objects of class <code>shrink</code> .

Data set contained in the shrink-package

<code>deepvein</code>	deep vein thrombosis study
-----------------------	----------------------------

**Author(s)**

Daniela Dunkler, Georg Heinze  
 Maintainer: <daniela.dunkler@meduniwien.ac.at>

**References**

Sauerbrei W (1999) The use of resampling methods to simplify regression models in medical statistics. *Applied Statistics* **48**(3): 313-329.  
 Verweij P, van Houwelingen J (1993) Cross-validation in survival analysis. *Statistics in Medicine* **12**(24): 2305-2314.

**See Also**

[shrink](#), [coef.shrink](#), [predict.shrink](#), [print.shrink](#)

**Examples**

```
# 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)
linpred <- intercept + x1*beta[1] + x2*beta[2]
prob <- exp(linpred)/(1 + exp(linpred))
runis <- runif(n,0,1)
ytest <- ifelse(runis < prob,1,0)
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
```

---

coef.shrink	<i>Returns Shrunk Regression Coefficients from Objects of Class shrink</i>
-------------	--

---

### Description

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

### Usage

```
## S3 method for class 'shrink'  
coef(object, ...)
```

### Arguments

object	object of class shrink.
...	further arguments.

### Value

A vector with shrunken regression coefficients.

### Author(s)

Daniela Dunkler, Georg Heinze

### See Also

[shrink](#), [print.shrink](#)

---

deepvein	<i>Deep Vein Thrombosis Study</i>
----------	-----------------------------------

---

### Description

A data frame containing time to recurrence of thrombosis and several potential prognostic factors for 929 individuals.

The data are a modified and partly simulated version of the data set used by Eichinger et al. (Eichinger S, Heinze G, Jandeck LM, Kyrle P (2010). Risk Assessment of Recurrence in Patients With Unprovoked Deep Vein Thrombosis or Pulmonary Embolism - The Vienna Prediction Model. *Circulation*, 121(14): 1630-1636.), and are available under a GPL-2 license.

**Usage**

```
data(deepvein)
```

**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.

**status** = 1 recurrence of thrombosis.

**sex** gender.

**fiimut** 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.

**References**

Eichinger S, Heinze G, Jandek LM, Kyrle P (2010). Risk Assessment of Recurrence in Patients With Unprovoked Deep Vein Thrombosis or Pulmonary Embolism - The Vienna Prediction Model. *Circulation*, **121**(14): 1630-1636.

**Examples**

```
data("deepvein")
summary(deepvein)

library("survival")
deepvein$loc <- relevel(deepvein$loc, ref = "distal")
fitfull <- coxph(Surv(time, status) ~ sex + fiimut + fvleid + log2ddim +
                bmi + durther + age + loc, data = deepvein, x = TRUE)
summary(fitfull)

shrink(fitfull, type = "global", method = "dfbeta")
shrink(fitfull, type = "parameterwise", method = "dfbeta")
shrink(fitfull, type = "parameterwise", method = "dfbeta",
       join = list(c("locproximal", "locPE")))

#shrink(fitfull, type = "global", method = "jackknife")
#shrink(fitfull, type = "parameterwise", method = "jackknife")
#shrink(fitfull, type = "parameterwise", method = "jackknife",
#       join = list(c("locproximal", "locPE")))
```

```
# backward elimination based on AIC
fitselect <- step(fitfull, direction = "backward")
summary(fitselect)

shrink(fitselect, type = "global", method = "dfbeta")
shrink(fitselect, type = "parameterwise", method = "dfbeta")
shrink(fitselect, type = "parameterwise", method = "dfbeta",
       join = list(c("locproximal", "locPE")))

#shrink(fitselect, type = "global", method = "jackknife")
#shrink(fitselect, type = "parameterwise", method = "jackknife")
#shrink(fitselect, type = "parameterwise", method = "jackknife",
#       join = list(c("locproximal", "locPE")))
```

---

predict.shrink	<i>Predict Method for Objects of Class shrink</i>
----------------	---

---

## Description

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

## Usage

```
## S3 method for class 'shrink'
predict(object, newdata = NULL, type = c("link", "response", "lp", "risk",
    "expected", "terms"), 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 used in the fit object.
type	the type of prediction required.
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, 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.

fit                    predictions.

**Author(s)**

Daniela Dunkler, Georg Heinze

**See Also**

[shrink](#)

**Examples**

```
library("mfp")
data("GBSG")
dat <- GBSG
set.seed(79)
dat <- dat[sample(seq_len(nrow(dat)), size = 400, replace = FALSE),]

fit1 <- mfp(Surv(rfst, cens) ~ fp(age, df = 4, select = 0.05) +
            fp(prm, df = 4, select = 0.05), family = cox, data = dat)

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

# unshrunk
plot(20:80, predict(fit1, newdata = data.frame(age = 20:80, prm = 0), type = "lp") -
     predict(fit1, newdata = data.frame(age = 50, prm = 0), type = "lp"), xlab = "Age",
     ylab = "Log hazard relative to 50 years", type = "l", lwd = 2)

# globally shrunk
lines(20:80, predict(dfbeta.global, newdata = data.frame(age = 20:80, prm = 0), type = "lp") -
      predict(dfbeta.global, newdata = data.frame(age = 50, prm = 0), type = "lp"),
      lty = 4, col = "blue", lwd = 2)

# jointly shrunk
lines(20:80, predict(dfbeta.join, newdata = data.frame(age = 20:80, prm = 0), type = "lp") -
      predict(dfbeta.join, newdata = data.frame(age = 50, prm = 0), type = "lp"),
      lty = 3, col = "red", lwd = 2)

# parameterwise shrunk
lines(20:80, predict(dfbeta.pw, newdata = data.frame(age = 20:80, prm = 0), type = "lp") -
      predict(dfbeta.pw, newdata = data.frame(age = 50, prm = 0), type = "lp"),
      lty = 2, col = "green", lwd = 2)

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

---

print.shrink	<i>Print Method for Objects of Class shrink</i>
--------------	---

---

### Description

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

### Usage

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

### Arguments

x	object of class shrink.
...	further arguments.

### Author(s)

Daniela Dunkler, Georg Heinze

### See Also

[shrink](#), [coef.shrink](#)

---

shrink	<i>Global, Parameterwise, and Joint Shrinkage of Regression Coefficients</i>
--------	--

---

### Description

Obtain global, parameterwise, and joint post-estimation shrinkage factors for regression coefficients from fit objects of class lm, class glm with family = c("gaussian", "binomial"), class coxph, or class mfp with family = c(cox, gaussian, binomial).

### Usage

```
shrink(fit, type = "parameterwise", method = "jackknife", join = NULL)
```



**Arguments**

<code>fit</code>	a fit object of class <code>lm</code> , <code>glm</code> , <code>coxph</code> or <code>mfp</code> . The fit object must have been called with <code>x = TRUE</code> (and <code>y = TRUE</code> in case of <code>lm</code> ).
<code>type</code>	of shrinkage, either "parameterwise" (or "p"; default) or "global" (or "g") shrinkage.
<code>method</code>	of shrinkage estimation, either "jackknife" (or "j"; default, based on leave-one-out resampling) or "dfbeta" (or "d"; excellent approximation based on DFBETA residuals).
<code>join</code>	compute optional joint shrinkage factors for sets of specified columns of the design matrix, if <code>type = "parameterwise"</code> . See details.

**Details**

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

Joint shrinkage factors may be useful when analysing highly correlated and/or semantically related columns of the design matrix, such as 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 such '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; e. g. `join = list(c("dummy1", "dummy2", "dummy3"), c("main1", "main2", "interaction"), c("varX.fp1", "varX.fp2"))`.

`shrink` also works for models incorporating restricted cubic splines computed with the `rcs` function from the `rms` library. A joint shrinkage factor of variable `varX` transformed with `rcs` can be obtained by `join = list(c("varX"))`.

For fit objects of class `coxph` or `glm` with `family = "binomial"` 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.

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`, with `type = "parameterwise"`, the shrunken regression coefficients can be directly estimated. This `postfit` is retained in the `$postfit` slot of the `shrink` object.

**Value**

`shrink` returns an object with the following components:

<code>shrinkage</code>	a vector of shrinkage factors of regression coefficients.
------------------------	---

<code>vcov.shrinkage</code>	a covariance matrix of shrinkage factors.
<code>shrunken</code>	a vector with the shrunken regression coefficients.
<code>postfit</code>	an optional postfit model with shrunken regression coefficients and associated standard errors if <code>type = "parameterwise"</code> and <code>join = NULL</code> .
<code>fit</code>	the original (unshrunken) fit object.
<code>type</code>	the requested shrinkage type.
<code>method</code>	the requested shrinkage method.
<code>call</code>	the function call.

**Note**

For fit objects of class `mfp` with `family = binomial` or `gaussian` the regression coefficients of `fit` (obtained by `coef(fit)`) and `fit$fit` (`coef(fit$fit)`) may not always be identical, because of `mfp`'s pretransformation applied to the explanatory variables in the model. The `shrink` function uses the regression coefficients from `fit$fit` which correspond to the pretransformed explanatory variables.

**Author(s)**

Daniela Dunkler, Georg Heinze

**References**

- Sauerbrei W (1999) The use of resampling methods to simplify regression models in medical statistics. *Applied Statistics* **48**(3): 313-329.
- Verweij P, van Houwelingen J (1993) Cross-validation in survival analysis. *Statistics in Medicine* **12**(24): 2305-2314.

**See Also**

[coef.shrink](#), [predict.shrink](#), [print.shrink](#)

**Examples**

```
# Example with mfp (family = cox)
library("mfp")
data("GBSG")
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
cov2cor(dfbeta.pw$vcov.shrinkage)
sqrt(diag(dfbeta.pw$vcov.shrinkage))

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")))

# Example with rcs
library("rms")
fit2 <- coxph(Surv(rfst, cens) ~ rcs(age) + rcs(prm), 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("age"), c("prm")))

# Examples with glm & mfp (family=binomial)
set.seed(888)
intercept <- 1
beta <- c(0.5, 1.2)
n <- 1000
x1 <- rnorm(n,1,1)
x2 <- rbinom(n, 1, 0.3)
linpred <- intercept + x1*beta[1] + x2*beta[2]
prob <- exp(linpred)/(1 + exp(linpred))
runis <- runif(n,0,1)
ytest <- ifelse(runis < prob,1,0)
simdat <- data.frame(cbind(y = ifelse(runis < prob, 1, 0), x1, x2))

fit2 <- glm(y ~ x1 + x2, family = binomial, data = simdat, x = TRUE)
summary(fit2)

shrink(fit2, type = "global", method = "dfbeta")
shrink(fit2, type = "parameterwise", method = "dfbeta")
shrink(fit2, 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)
Pima$type2 <- as.numeric(Pima$type)-1
fit3 <- mfp(type2 ~ npreg + glu + bmi + ped + fp(age, select = 0.05),
           family = binomial, data = Pima)
fit3

shrink(fit3, type = "global", method = "dfbeta")
shrink(fit3, type = "parameterwise", method="dfbeta")

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

```

```
fit4 <- glm(Postwt ~ Prewt + Treat, family = gaussian, data = anorexia, x = TRUE)
fit4

shrink(fit4, type = "global", method = "dfbeta")
shrink(fit4, type = "parameterwise", method = "dfbeta")
shrink(fit4, type = "parameterwise", method = "dfbeta",
       join = list(c("Treat1", "Treat3")))

fit5 <- lm(Postwt ~ Prewt + Treat, data = anorexia, x = TRUE, y = TRUE)
fit5

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

utils::data("GAGurine", package="MASS")
fit6 <- mfp(Age ~ fp(GAG, select = 0.05), family = gaussian, data = GAGurine)
fit6

shrink(fit6, type = "global", method = "dfbeta")
shrink(fit6, type = "parameterwise", method = "dfbeta")
shrink(fit6, type = "parameterwise", method = "dfbeta",
       join = list(c("GAG.1", "GAG.2")))
```

# Index

- \*Topic **datasets**
  - deepvein, [4](#)
- \*Topic **models**
  - shrink, [8](#)
  - shrink-package, [2](#)
- \*Topic **nonlinear**
  - shrink, [8](#)
  - shrink-package, [2](#)
- \*Topic **package**
  - shrink-package, [2](#)
- \*Topic **regression**
  - shrink, [8](#)
  - shrink-package, [2](#)
- \*Topic **survival**
  - shrink, [8](#)
  - shrink-package, [2](#)
- \*Topic **utilities**
  - coef.shrink, [4](#)
  - predict.shrink, [6](#)
  - print.shrink, [8](#)

coef.shrink, [3](#), [4](#), [8](#), [10](#)

deepvein, [4](#)

predict.shrink, [3](#), [6](#), [10](#)

print.shrink, [3](#), [4](#), [8](#), [10](#)

shrink, [3](#), [4](#), [7](#), [8](#), [8](#)

shrink-package, [2](#)