Package 'robreg3S'

October 14, 2022

Type Package

Title Three-Step Regression and Inference for Cellwise and Casewise Contamination

Version 0.3

Date 2015-12-28

Author Andy Leung, Hongyang Zhang, Ruben Zamar

Maintainer Andy Leung <andy.leung@stat.ubc.ca>

Description Three-step regression and inference for cellwise and casewise contamination.

Depends GSE, MASS

Imports robustbase

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2015-12-30 08:36:45

R topics documented:

`robreg3S` *Robust regression estimation and inference in the presence of cellwise and casewise contamination*

Description

Finds 3S-robust regression estimator using the adaptive consistent filter.

Usage

```
robreg3S(y, x, dummies=NULL, filter=TRUE, alpha=0.20, K=5, ...)
```

Arguments

y	vector of responses.
x	matrix of the numerical variables.
dummies	matrix of the dummy covariates, i.e., where each column are 0–1 vectors.
filter	logical, whether the filtering is used. Default value is TRUE.
alpha	1-alpha upper quantile (and alpha lower quantile) of the covariate distribution used in tail comparison in the first step. An exponential tail is used as the reference distribution. Default value is 0.20.
K	number of alternating M-S iterations in the estimation of the coefficients of the dummy covariates. Default value is 5. See Leung et al. for more details.
...	optional arguments to be used in the computation of GSE in the second step. See GSE

Details

This function computes 3S-robust regression as described in Leung et al. (2015).

If the model contains dummy variables (i.e., `dummies != NULL`), 3S-regression is computed using an iterative algorithm as described in Leung et al. (2015). Briefly, the algorithm first estimates the coefficients of the dummies using an M-estimator of regression and the coefficients of the continuous covariates using the original 3S-regression. See Leung et al. (2015) for more details.

Value

A list with components:

Summary.Table	Matrix of information available about the estimator. It contains regression coefficients, and for `dummies != NULL`, columns for the standard error, t-statistic, and p-value.
coef	vector of regression coefficients.
acov	matrix of the asymptotic covariate matrix, only for `dummies != NULL`.
resid	vector of residuals, that is the response minus the fitted values.
sigma.hat	the estimated residual standard error.
MD	the squared Mahalanobis distances of each observation based on the continuous covariates to the generalized location S-estimator with respect to the generalized scatter S-estimator.
xfilter	filtered matrix of the numerical variables from Step 1 of the estimator.
ximpute	matrix of the numerical variables with filtered cells imputed from Step 2 of the estimator.
weight	vector of the weights used in the estimation of the location generalized S-estimator. Not meant to be accessed.
Syx	estimated generalized S-scatter from Step 2. Not meant to be accessed.
myx	estimated generalized S-location from Step 2. Not meant to be accessed.

Author(s)

Andy Leung <andy.leung@stat.ubc.ca>, Hongyang Zhang, Ruben H. Zamar

References

Leung, A. , Zamar, R.H., and Zhang, H. Robust regression estimation and inference in the presence of cellwise and casewise contamination. arXiv:1509.02564.

See Also

GSE, generate.cellcontam.regress, generate.casecontam.regress, generate.casecontam.regress.dummies, generate.casecontam.regress.dummies

Examples

```
## Boston housing data
data(Boston, package="MASS")
boston <- Boston; rm(Boston)
boston$crim <- log(boston$crim)
boston$nox <- boston$nox^2
boston$rm <- boston$rm^2
boston$dis <- log(boston$dis)
boston$lstat <- log(boston$lstat)
boston$medv <- log(boston$medv)
boston$black <- boston$black/1000
boston$age <- boston$age/100
boston$tax <- boston$tax/100
boston$indus <- boston$indus/100
boston <- subset( boston, select=c(medv, crim, nox, rm, age, dis, tax, ptratio, black, lstat) )

## LS, MM, 3S
set.seed(100)
fit.LS <- lm(medv ~  ., data=boston)
fit.MM <- robustbase::lmrob(medv ~  ., data=boston)
fit.2S <- robreg3S( y=boston$medv, x=as.matrix(subset(boston,select=-medv)), filter = FALSE )
fit.3S <- robreg3S( y=boston$medv, x=as.matrix(subset(boston,select=-medv)) )

## Compare estimated coefficients
nrow(boston) *sum(( coef(fit.LS)[-1] - coef(fit.3S)[-1])^2* apply(boston[,-1], 2, mad)^2)
nrow(boston) *sum(( coef(fit.MM)[-1] - coef(fit.3S)[-1])^2* apply(boston[,-1], 2, mad)^2)
nrow(boston) *sum(( coef(fit.2S)[-1] - coef(fit.3S)[-1])^2* apply(boston[,-1], 2, mad)^2)

## Summary table
summary(fit.3S)
```

simulation-tools *Data generator for simulation study on cell- and case-wise contamination*

Description

Includes the data generator for the simulation study on cell- and case-wise contamination that appears on Leung et al. (2015).

Usage

```
generate.randbeta(p)

generate.cellcontam.regress(n, p, A, sigma, b, k, cp)

generate.casecontam.regress(n, p, A, sigma, b, l, k, cp)

generate.cellcontam.regress.dummies(n, p, pd, probd, A, sigma, b, k, cp)

generate.casecontam.regress.dummies(n, p, pd, probd, A, sigma, b, l, k, cp)
```

Arguments

n	integer indicating the number of observations to be generated.
p	integer indicating the number of continuous variables to be generated.
pd	integer indicating the number of dummy variables to be generated.
probd	vector of quantiles of length pd. To generate dummy variables pd continuous variables are first generated. Then, the variables are dichotomize at normal quantiles of probd.
A	a correlation matrix. See also generate.randcorr.
sigma	residual standard deviation.
b	vector of regression coefficients.
k	size of cellwise outliers and vertical outliers. See Leung et al. for details.
l	size of leverage outliers. See Leung et al. for details.
cp	proportion of cell- or case-wise contamination. Maximum of 10% for cellwise and 50% for casewise.

Value

A list with components:

x	multivariate normal sample with cell- or case-wise contamination.
y	vector of responses.
dummies	vector of dummies.

Author(s)

Andy Leung <andy.leung@stat.ubc.ca>, Hongyang Zhang, Ruben H. Zamar

References

Leung, A. , Zamar, R.H., and Zhang, H. Robust regression estimation and inference in the presence of cellwise and casewise contamination. arXiv:1509.02564.

See Also

generate.randcorr

Examples

```
################################################
## Cellwise contaminated data simulation
## (continuous covariates only)
set.seed(10)
b <- 10*generate.randbeta(p=15)
A <- generate.randcorr(cond=100, p=15)
dat <- generate.cellcontam.regress(n=300, p=15, A=A, sigma=0.5, b=b, k=10, cp=0.05)

## LS
fit.LS <- lm( y ~ x, dat)
mean((coef(fit.LS)[-1] - b)^2)

## MM regression
fit.MM <- robustbase::lmrob( y ~ x, dat)
mean((coef(fit.MM)[-1] - b)^2)

## 3S regression
fit.3S <- robreg3S( y=dat$y, x=dat$x, init="imputed")
mean((coef(fit.3S)[-1] - b)^2)



################################################
## Casewise contaminated data simulation
## (continuous covariates only)
set.seed(10)
b <- 10*generate.randbeta(p=10)
A <- generate.randcorr(cond=100, p=10)
dat <- generate.casecontam.regress(n=200, p=10, A=A, sigma=0.5, b=b, l=8, k=10, cp=0.10)

## LS
fit.LS <- lm( y ~ x, dat)
mean((coef(fit.LS)[-1] - b)^2)

## MM regression
fit.MM <- robustbase::lmrob( y ~ x, dat)
mean((coef(fit.MM)[-1] - b)^2)
```

```
## 3S regression
fit.3S <- robreg3S( y=dat$y, x=dat$x, init="imputed")
mean((coef(fit.3S)[-1] - b)^2)


## Not run:
#################################################
## Cellwise contaminated data simulation
## (continuous and dummies covariates)
set.seed(10)
b <- 10*generate.randbeta(p=15)
A <- generate.randcorr(cond=100, p=15)
dat <- generate.cellcontam.regress.dummies(n=300, p=12, pd=3,
   probd=c(1/2,1/3,1/4), A=A, sigma=0.5, b=b, k=10, cp=0.05)

## LS
fit.LS <- lm( dat$y ~ dat$x + dat$dummies)
mean((coef(fit.LS)[-1] - b)^2)

## MM regression
fit.MM <- robustbase::lmrob( dat$y ~ dat$x + dat$dummies)
mean((coef(fit.MM)[-1] - b)^2)

## 3S regression
fit.3S <- robreg3S( y=dat$y, x=dat$x, dummies=dat$dummies, init="imputed")
mean((coef(fit.3S)[-1] - b)^2)


#################################################
## Casewise contaminated data simulation
## (continuous and dummies covariates)
set.seed(10)
b <- 10*generate.randbeta(p=15)
A <- generate.randcorr(cond=100, p=15)
dat <- generate.casecontam.regress.dummies(n=300, p=12, pd=3,
   probd=c(1/2,1/3,1/4), A=A, sigma=0.5, b=b, l=7, k=10, cp=0.10)

## LS
fit.LS <- lm( dat$y ~ dat$x + dat$dummies)
mean((coef(fit.LS)[-1] - b)^2)

## MM regression
fit.MM <- robustbase::lmrob( dat$y ~ dat$x + dat$dummies)
mean((coef(fit.MM)[-1] - b)^2)

## 3S regression
fit.3S <- robreg3S( y=dat$y, x=dat$x, dummies=dat$dummies, init="imputed")
mean((coef(fit.3S)[-1] - b)^2)


## End(Not run)
```

Index