Package ‘gamreg’

November 17, 2017

Type Package
Title Robust and Sparse Regression via Gamma-Divergence
Version 0.3
Date 2017-11-17
Author Takayuki Kawashima
Maintainer Takayuki Kawashima <t-kawa@ism.ac.jp>
Description Robust regression via gamma-divergence with L1, elastic net and ridge.
License GPL (>= 2)
LazyData TRUE
Suggests mvtnorm
Imports glmnet, robustHD, foreach, doParallel
LinkingTo Rcpp, RcppArmadillo
RoxygenNote 5.0.1
NeedsCompilation yes
Repository CRAN
Date/Publication 2017-11-17 08:35:34 UTC

R topics documented:
cv.gam ................................................................. 2
Index ......................................................... 4
Robust Cross-Validation

Description

Compute Robust Cross-Validation for selecting best model.

Usage

```r
cv.gam(x, y, init.mode = c("sLTS", "RLARS", "RANSAC"),
    lambda.mode = "lambda0", lmax = 1, lmin = 0.05, nlambda = 50,
    fold = 10, ncores = 1, gam = 0.1, gam0 = 0.5, intercept = "TRUE",
    alpha = 1, ini.subsamp = 0.2, ini.cand = 1000, alpha.LTS = 0.75,
    nlambda.LTS = 40)
```

Arguments

- **X**: Predictor variables Matrix.
- **Y**: Response variables Matrix.
- **init.mode**: "sLTS": a initial point is the estimate of sparse least trimmed squares. "RLARS": a initial point is the estimate of Robust LARS. "RANSAC": a initial point is the estimate of RANSAC algorithm.
- **lambda.mode**: "lambda0": Robust Cross-Validation uses grids on range [0.05lambda0,lambda0] with log scale, where lambda0 is an estimator of sparse tuning parameter which would shrink regression coefficients to zero.
- **lmax**: When lambda.mode is not lambda0, upper bound of range of grids is lmax.
- **lmin**: When lambda.mode is not lambda0, lower bound of range of grids is lmin.
- **nlambda**: The number of grids for Robust Cross-Validation.
- **fold**: the number of folds for K-fold Robust Cross-Validation. If fold equals to sample size, Robust Cross-Validation is leave-one-out method.
- **ncores**: positive integer giving the number of processor cores to be used for parallel computing (the default is 1 for no parallelization).
- **gam**: Robust tuning parameter of gamma-divergence for regression.
- **gam0**: tuning parameter of Robust Cross-Validation.
- **intercept**: Should intercept be fitted TRUE or set to zero FALSE
- **alpha**: The elasticnet mixing parameter, with 0 ≤ α ≤ 1. alpha=1 is the lasso penalty, and alpha=0 is the ridge penalty.
- **ini.subsamp**: The fraction of subsamples in "RANSAC".
- **ini.cand**: The number of candidates for estimating initial points in "RANSAC".
- **alpha.LTS**: The fraction of subsamples for trimmed squares in "sLTS".
- **nlambda.LTS**: The number of grids for sparse tuning parameter in "sLTS".
Details

If the "RANSAC" is used as the initial point, the parameter ini.subsamp and ini.cand can be determined carefully. The smaller ini.subsamp is, the more robust initial point is. However, less efficiency.

Value

| lambda | A numeric vector giving the values of the penalty parameter. |
| fit    | All results at each lambda.                                  |
| Rocv   | The result of best model by Robust Cross-Validation.         |

Author(s)

Takayuki Kawashima

References


Examples

```r
## generate data
library(mvtnorm)
# number of observations
n <- 30
# number of explanatory variables
p <- 10
# contamination ratio
epsilon <- 0.1
# intercept
beta0 <- 0.0
# regression coefficients
beta <- c(numeric(p))
beta[1] <- 1
beta[4] <- 4

Sigma <- 0.2^t(sapply(1:p, function(i, j) abs(i-j), 1:p))
X <- rmvnorm(n, sigma=Sigma)  # explanatory variables
e <- rnorm(n)  # error terms

i <- 1:ceiling(epsilon*n)  # index of outliers
e[i] <- e[i] + 20  # vertical outliers

Y <- beta0*(numeric(n)+1) + X%*%beta

res <- cv.gam(X, Y, nlambda = 5, nlambda.LTS=20, init.mode="sLTS")
```
Index

cv.gam, 2