Package ‘GRPtests’

October 11, 2019

Type Package

Title Goodness-of-Fit Tests in High-Dimensional GLMs

Version 0.1.0

Date 2019-09-14

Author Jana Jankova [aut, cre], Rajen Shah [aut], Peter Buehlmann [aut], Richard Samworth [aut]

Maintainer Jana Jankova <jana.jankova@gmail.com>

Description Methodology for testing nonlinearity in the conditional mean function in low- or high-dimensional generalized linear models, and the significance of (potentially large) groups of predictors. Details on the algorithms can be found in the paper by Jankova, Shah, Buehlmann and Samworth (2019) <arXiv:1908.03606>.

License GPL

Imports glmnet, randomForest, MASS, stats, RPtests

Suggests xyz

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2019-10-11 12:00:04 UTC

R topics documented:

GRPgrouptest *Test significance of groups or individual predictors in high-dimensional generalized linear models*

Description

The function can test significance of (potentially large) groups of predictors in low- and high-dimensional generalized linear models. Outputs a p-value.

Usage

```
GRPgrouptest(X, y, fam = c("gaussian", "binomial", "poisson"), G,
  B = 1000L, penalize = ifelse(p - length(G) >= n, TRUE, FALSE))
```

Arguments

`X` Input matrix with `n` rows, each a `p`-dimensional observation vector.

`y` Response vector.

`fam` Must be "gaussian", "binomial" or "poisson".

`G` A vector with indices of variables, whose significance we wish to ascertain, after controlling for variables in `X`. The size of `G` can be at most `p-2`.

`B` The number of bootstrap samples to approximate the distribution of the test statistic. Note that the p-value returned will always be at least `1/(B+1)`.

`penalize` If `TRUE`, penalization is used when fitting GLM models.

Details

The function can test the significance of a set of variables in a generalized linear model, whose indices are specified by `G`. `penalize = TRUE` is needed for high-dimensional settings where the number of variables not in `G` is larger than the number of observations. We then employ a penalized regression to regress `y` on to these variables implemented in `cv.glmnet` from package `glmnet`. For the low-dimensional case, an unpenalized regression may be used.

Value

The output is a single p-value.

References

Janková, J., Shah, R. D., Bühlmann, P. and Samworth, R. (2019) *Goodness-of-fit testing in high-dimensional generalized linear models* `https://arxiv.org/abs/1908.03606`

Examples

```
# Testing significance of a group of predictors in logistic regression

set.seed(1)
X <- matrix(rnorm(300*50), 300, 50)
z <- X[, 1:5] %*% rep(1, 5)
pr <- 1/(1 + exp(-z))
y <- rbinom(nrow(X), 1, pr)
(out <- GRPgrouptest(X, y, fam = "binomial", G = 5:10, B = 1000))
```

`GRPtest` *Goodness-of-fit test for high-dimensional generalized linear models*

Description

The function can test goodness-of-fit of a low- or high-dimensional generalized linear model (GLM) by detecting the presence of nonlinearity in the conditional mean function of y given X. Outputs a p-value.

Usage

```
GRPtest(X, y, fam = c("gaussian", "binomial", "poisson"),
  RP_function = NULL, nsplits = 5L, penalize = ifelse(p >=
  floor(n/2), TRUE, FALSE), output_all = FALSE)
```

Arguments

`X` Input matrix with `n` rows, each a `p`-dimensional observation vector.

`y` Response vector.

`fam` Must be "gaussian", "binomial" or "poisson".

`RP_function` (optional) User specified function for residual prediction (see Details below).

`nsplits` Number of splits of the data set (see Details below).

`penalize` `TRUE` if penalization should be used when fitting the GLM models (see Details below).

`output_all` If `TRUE`, outputs all p-values from `nspilts` splits of the data.

Details

This function tests if the conditional mean of `y` given `X` could be originating from a GLM family specified by the user via `fam`.

The function works by splitting the data into parts A and B, and computes a GLM fit on both parts. If `penalize` == `TRUE`, these fits use `cv.glmnet` from package `glmnet`, otherwise they use `glmnet` with penalty set to 0. If `RP_function` (optional) is not supplied by the user, `randomForest` is used to predict remaining signal from the residuals from GLM fit on part A. The test statistic is proportional to the dot product between the random forest prediction and residuals from GLM fit on part B. If nsplits is greater than one, the above procedure is repeated nsplits times and the resulting p-values are aggregated using the approach from Meinshausen at al. (2012)

A user may supply their own residual prediction function to replace random forest via parameter RP_function (see Examples for use). The function must take as arguments an input matrix XA, vector resA (with length nrow(XA)) and matrix XB. Its role is to regress resA on input matrix XA with a preferred residual prediction method and output a vector with dimensions nrow(XB) that contains predictions of this fit on input XB.

Value

If output_all = FALSE, the function outputs a single p-value. Otherwise it returns a list containing the aggregated p-value in pval and a vector of p-values from all splits in pvals.

References

Janková, J., Shah, R. D., Bühlmann, P. and Samworth, R. (2019) *Goodness-of-fit testing in high-dimensional generalized linear models* https://arxiv.org/abs/1908.03606 Meinshausen, N., Meier, L. and Bühlmann, P. (2012) *p-Values for High-Dimensional Regression* Journal of the American Statistical Association, 104:488, 1671-1681

Examples

```
# Testing for nonlinearity: Logistic link function

set.seed(1)
X <- matrix(rnorm(300*30), 300, 30)
z <- X[, 1] + X[, 2]^4
pr <- 1/(1 + exp(-z))
y <- rbinom(nrow(X), 1, pr)
(out <- GRPtest(X, y, fam = "binomial", nsplits = 5))

# Testing for nonlinearity: Define your own RP function
# use package xyz

my_RP_function <- function(XA, resA, XB){
  xyz_fit <- xyz_regression(XA, resA)
  predict(xyz_fit, newdata = as.matrix(XB))[,5]
}

library(xyz)
set.seed(2)
X <- matrix(rnorm(500*30), 500, 30)
z <- X[,1:3]%*%rep(1,3) + 1*X[, 1]*X[,5]
mu <- exp(z)
y <- rpois(n = nrow(X), lambda = mu)
(out <- GRPtest(X, y, fam = "poisson", RP_function = my_RP_function))
```

Index