'BOLTSSIRR' Package for BOLT-SSI: Fully Screening Interaction Effects for Ultra-High Dimensional Data

1 Overview

This vignette provides an introduction to the 'BOLTSSIRR' package. BOLTSSI is a statistical approach for detecting interaction effects among predict variables to response variables, which is often an crucial step in regression modeling of real data for various applications. Through this publicly available package, we provide a unified environment to carry out interaction pursuit using a simple sure screening procedure (SSI) to fully detect significant pure interactions between predict variables and the response variable in the high or ultra-high dimensional generalized linear regression models. Furthermore, we suggest to discretize continuous predict variables, and utilize the Boolean operation for the marginal likelihood estimates. The so-called 'BOLTSSI' procedure is proposed to accelerate the sure screening speed of the procedure.

After screening the interaction effects, we just use penalzied likelihood function with LASSO penalty to further select the variables including the interaction terms. The objective function is

$$-loglik/nobs + \lambda * penalty.$$

This vignette is organized as follows. Section 2 introduces the basic principle of our methods. Section 3 illustrates how to screen the inteaction effecs and choose the variables by using this package.

2 Introduction

2.1 SSI

Assume that given the predictor vector x, the conditional distribution of the random variable Y belongs to an exponential family, whose probability density function has the canonical form

$$f_{Y|x}(y|x) = \exp\{y\theta(x) - b(\theta(x)) + c(y)\}$$

where $b(\cdot)$ and $c(\cdot)$ are some known functions and $\theta(x)$ is a canonical natural parameter. Here we ignore the dispersion parameter ϕ in the above density function, since we only concentrate on the estimation of mean regression function. It is well known that the distributions in the exponential family include the Binomial, Gaussian, Gamma, Inverse-Gaussian and Poisson distributions.

We consider the following generalized linear model with two-way interaction:

$$E(Y|X = x) = b'(\theta(x)) = g^{-1}\left(\beta_0 + \sum_{i=1}^{p}\beta_i X_i + \sum_{i<j}\beta_{ij}X_iX_j\right)$$

for some link function $g(\cdot)$. And we focus on the canonical link function, hence $g^{-1}(\cdot) = b'$ and

$$\theta(x) = \beta_0 + \sum_{i=1}^{p}\beta_i X_i + \sum_{i<j}\beta_{ij}X_iX_j \triangleq \beta_0 + \sum_{i=1}^{p}\beta_i X_i + \sum_{i<j}\beta_{ij}X_{ij}.$$

Our method is based on the marginal likelihood ratio screening, which builds on the difference between two marginal negative log-likelihood functions in the following two models:

$$E(Y|X_i, X_j) = \beta_{i,j0} + \beta_{i,}X_i + \beta_{j,}X_j$$

and

$$E(Y|X_i, X_j) = \beta_{i,j0} + \beta_{i,}X_i + \beta_{j,}X_j + \beta_{ij}X_{ij}.$$

To ease the presentation, through this vignette, X_{ij} is referred to as an important interaction if its regression coefficient β_{ij} is nonzero. X_i is called an active interaction variable if there exists some $1 \leq i \neq j \leq p$ such that X_{ij} is an important interaction. Our method "SSI" is based on the above difference and to implemnet the screening procedure for interaction terms.

2.2 BOLTSSI

Furthermore, the "BOLTSSI" procedure is proposed to accelerate the sure screening speed of the procedure. The first step of BOLT-SSI is to discretize one continuous attribute by creating one categorical variabe wtih a specified number of levels. After discretization, the Boolean operation can be used to speedup SSI procedure, especially the algorithm to calculate the difference in the above section 2.1. All details can be seen in the paper (et al. [2018]).

3 Screening and Prediction

The package can be loaded with the command:

```
library(BOLTSSIRR)
```

3.1 The function 'BOLT_SSI'

Description

This function implements the Sure Independence Screening SSI and BOLTSSI for interaction screening.

Usage

```
BOLT_SSI(X,y,extra_pairs,code_num,thread_num)
```

Arguments

Arguments	
X	The design matrix, of dimensions $n * p$, without an intercept. Each row is an observation vector.
y	The response vector of dimension $n * 1$.
extra_pairs	The number of remaining interaction effects by using SSI or BOLTSSI,default is n.
code_num	The level of preditor varibales after discretization, default is 3.
thread_num	The number of thread_num, default is 4.

Value

Returns a matrix with three columns.

Values	
Th frist column	The value i, where i is the index of the avtive interaction variable X_i.
The second column	The value j, where j is the index of the avtive interaction variable X_j.
The third column	The difference between two marginal negative log-likelihood functions.

Examples

```
library(BOLTSSIRR)
set.seed(0)
p=300;n=100;rho=0.5
H<-abs(outer(1:p,1:p,"-"))
covxx=rho^H
cholcov = chol(covxx)
x0 = matrix(rnorm(n*p), n, p)
x = x0%*%cholcov

#gaussian response
set.seed(0)
y=2*x[,1]+2*x[,8]+3*x[,1]*x[,8]+rnorm(n)
model1=BOLT_SSI(x,y)
head(model1)

#binary response
set.seed(40)
feta = 2*x[,1]+2*x[,8]+3*x[,1]*x[,8];
fprob = exp(feta)/(1+exp(feta))
y = rbinom(n, 1, fprob)
model2=BOLT_SSI(x,y)
head(model2)
```

3.2 The function 'CV_BOLT_SSI_RR'

Description

This function fits a generalized linear model via penalized maximum likelihood. The regularization path is computed for the lasso penalty at a grid of values for the regularization parameter lambda. And it implements Cross-Validation for BOLTSSI. It can deal with all shapes of data, including very large sparse data matrices.

Usage

```
CV_BOLT_SSI_RR(X,y,extra_pairs,cod_num,nfold,nLambda,thread_num)
```

Arguments

Arguments	
X	The design matrix, of dimensions $n * p$, without an intercept. Each row is an observation vector.

Arguments	
y	The response vector of dimension $n * 1$.
extra_pairs	The number of remaining interaction effects by using SSI or BOLTSSI,default is n.
code_num	The level of preditor varibales after discretization, default is 3.
nfold	The number of folds, default is 10.
nLambda	The number of lambda, default is 100.
thread_num	The number of thread_num, default is 4.

Value

An object of class "CV_BOLTSSI_RR" is returned, which is a list with the ingredients of the crossvalidation fit.

Values	
lambdas	The values of lambda used in the fits.
beta	A nvars x length(lambda) matrix of coefficients, stored in sparse column format ("CsparseMatrix").
lambda_min	The value of lambda when predicition error arrives at its minimum by Cross-Validation
index_min	The index of optimal value
covs	The intercept vector of dimension $n * 1$.
pairs	A vector of all indexes of interaction effects when lambda is condisered as it optimal value.

Examples

```
set.seed(0)
p=300;n=100
rho=0.5
H<-abs(outer(1:p,1:p,"-"))
covxx=rho^H
cholcov = chol(covxx)

x0 = matrix(rnorm(n*p), n, p)
x = x0%*%cholcov
#gaussian response
set.seed(0)
y=2*x[,1]+2*x[,8]+3*x[,1]*x[,8]+rnorm(n)
model3=CV_BOLT_SSI_RR(x,y,extra_pairs=p,nfold=5)


Lambdas=model3$lambdas
Beta=model3$beta
Lambda.min=model3$lambda_min
index.min=which(Lambdas==Lambda.min)
Pairs=t(matrix(model3$pairs,nrow = 2))

m=length(final_Beta)
main_index=which(final_Beta[1:p]!=0)
inter_index=which(final_Beta[(p+1):m]!=0)
```

```
final_inter_index=Pairs[inter_index,]
main_index
final_inter_index
```

3.3 The function 'BOLT_Predict'

Description

This function makes predictions from a cross-validated BOLTSSIRR model, using the stored object, and the optimal value chosed for lambda.

Usage

```
BOLT_Predict(x,fit)
```

Arguments

Arguments	
x	Matrix of new values for x at which predcitons ate to be made. Must be a matrix.
fit	Fitted "CV_BOLT_SSI_RR" object.

Value

Returns a predicted response vector .

Examples

```
newx=matrix(rnorm(30*p),30)
yhat=BOLT_Predict(newx,model3)
```