Package 'SuRF.vs'

January 20, 2025

Title Subsampling Ranking Forward Selection (SuRF)

Version 1.1.0.1

Maintainer Toby Kenney <tkenney@mathstat.dal.ca>

Depends R (>= 3.2.3)

Imports glmnet, survival, dplyr

Suggests foreach, parallel, doParallel, knitr

Author Lihui Liu [aut],
Toby Kenney [aut, cre]

Description Performs variable selection based on subsampling, ranking forward selection. Details of the method are published in Lihui Liu, Hong Gu, Johan Van Limbergen, Toby Kenney (2020) SuRF: A new method for sparse variable selection, with application in microbiome data analysis Statistics in Medicine 40 897-919 <doi:10.1002/sim.8809>. Xo is the matrix of predictor variables. y is the response variable. Currently only binary responses using logistic regression are supported. X is a matrix of additional predictors which should be scaled to have sum 1 prior to analysis. fold is the number of folds for cross-validation. Alpha is the parameter for the elastic net method used in the subsampling procedure: the default value of 1 corresponds to LASSO. prop is the proportion of variables to remove in the each subsample. weights indicates whether observations should be weighted by class size. When the class sizes are unbalanced, weighting observations can improve results. B is the number of subsamples to use for ranking the variables. C is the number of permutations to use for estimating the critical value of the null distribution. If the 'doParallel' package is installed, the function can be run in parallel by setting ncores to the number of threads to use. If the default value of 1 is used, or if the 'doParallel' package is not installed, the function does not run in parallel. display.progress indicates whether the function should display messages indicating its progress. family is a family variable for the glm() fitting. Note that the 'glmnet' package does not permit the use of non-standard link functions, so will always use the default link function. However, the glm() fitting will use the specified link. The default is binomial with logistic regression, because this is a common use case. pval is the p-value for inclusion of a variable in the model. Under the null case, the number of false positives will be geometrically distributed with this as probability of success, so if this parameter is set to p, the expected number of false positives should be p/(1-p).

Encoding UTF-8

License GPL-3

VignetteBuilder knitr
NeedsCompilation no
Repository CRAN
Date/Publication 2022-01-08 01:52:49 UTC

Contents

`dataclean` *dataclean*

Description

This function is to 1)Scale the count data (count data only) to proportion 2)create a data frame consisting of proportion data, and 3) Keep an variable name list (original variable names and names in terms of X's, e.g.X1,X2,..,etc.) #environmental data (host genome and other information about observations)

Usage

```
dataclean(X.c, X.o, y)
```

Arguments

`X.c`	data frame that has count data from all levels (only count data will be row scaled)
`X.o`	data frame that has other environmental variables (no scaling will be done, those variables will scaled together with proportion data in LASSO step)
`y`	a vector representing the outcome (0 or 1 for binomial model)

Value

data.Xy: a dataframe containing all variables named as X1,X2,...,Xp and the binary outcome (called status)in the last column; this data frame will be used in the other functions for data analysis

Ranking *Ranking*

Description

This function is to rank the variables after B subsampings;It also removes the highly correlated variables from lower level

Usage

```
Ranking(data, model)
```

Arguments

data	the data object return from the dataclean function (the last column is the outcome)
model	model object from sub-sampling procedure

Value

table: a table shows the ranked variable list with its frequency (descending order)

Beta: coefficients flag (1 or 0) indicating if the variable is selected; intercept is not included;

Ranking_cox *Ranking_cox*

Description

This function is to rank the variables after B subsampings for cox proportional model; It also removes the highly correlated variables from lower level

Usage

```
Ranking_cox(data, model)
```

Arguments

data	the data object return from the dataclean function(the last column is the outcome)
model	cox proportion model object from sub-sampling procedure (B times)

Value

table: a table shows the ranked variable list with its frequency (descending order)

Beta: coefficients flag (1 or 0) indicating if the variable is selected; intercept is not included;

selectnew *selectnew*

Description

This function is to add new node (new deviance distribution for adding the new variable)

Usage

```
selectnew(
  vslist,
  ranktable,
  data,
  weights,
  ncores = 1,
  family,
  alpha_l,
  alpha_u,
  C
)
```

Arguments

ranktable	:ranking table from ranking step
family	:generalized model families
alpha_l	is the minimum significance level(>=0)
data	the dataframe should be arranged in the way such that columns are X1,X2,X3....,Xp, status. Where Xi's are variables and status is the outcome(for the logistic regression, the outcome is in terms of 0/1)
ncores	no of parallel computing cores
C	the number of permutation times
alpha_u	the upper significance level
weights	In a binomial model, weights: =TRUE: if weighted version is desired; =FALSE, otherwise ; In other models,weights: =vector of weights of the same size as the sample size N: if weighted version is desired;=FALSE, otherwise (other generalized model)
vslist	the current list of selected variables

Value

vslist: the updated list of selected variables

dev.dist: deviance distributions used for selecting the new variable

vtlist has 1)alpha.range for the newly selected variable,2)selvar: the newly selected variable name,3)pval: pvalue for the newly selected varaible,and 4)dev: the deviance value contributed by the newly selected variable

selpath *selpath*

Description

#This function is to trace the selection path

Usage

```
selpath(data, weights, ranktable, ncores, family, C, alpha_u)
```

Arguments

ranktable	:ranking table from ranking step
family	:generalized model families
data	the dataframe should be arranged in the way such that columns are X1,X2,X3....,Xp, status. Where Xi's are variables and status is the outcome(for the logistic regression, the outcome is 0/1)
ncores	no of parallel computing cores
C	the number of permutation times
alpha_u	the upper significance level
weights	In a binomial model, weights: =TRUE: if weighted version is desired; =FALSE, otherwise ; In other models,weights: =vector of weights of the same size as the sample size N: if weighted version is desired;=FALSE, otherwise (other generalized model)

Value

selpoint: a list. it contains each selected variable point,information includes 1)vslist: the variable sect before selecting this variable listed in 'selvar' 2)alpha.range: the variable will be selected within this alpha range 3)pval: pvalue of the variable 4)selvar:selected variable 5)vslist:variable sect after selecting the variable listed in 'selvar'

sel.nodes: a list. deviance distributions used for selecting the new variable; it includes 1)vslist: the variable sect before the new selection 2)dev.dist: the permutation for selecting the new variable 3)vtlist that has i)pval: pvalue of the proposed variable ii)selvar: selected variable (the proposed variable is NULL if not selected) iii)dev:deviance contributed by the proposed variable

`selvar_alpha` *selvar_alpha*

Description

This function is to extract summarize the results at 'alpha' level from 'mod' object to obtain 1)final selected variables 'selvar' #2)pvalue of each selected variable according to the variables in the 'selvar' 3)deviance contributed by each selected variable (given the previous selected variables 4)deviance permutation distribution 5)cutoff value based on (1-alpha)percentile of the deviance permutation distribution;when no variable is selected, only return the last deviance distribution and the cutoff value;this function can be used separately after running selpath(); the alpha value must be >0 and <= alpha_u parameter from SURF()

Usage

```
selvar_alpha(res, alpha)
```

Arguments

`res`	'mod' object returned from 'selpath' function
`alpha`	alpha level(default alpha=0.05)(a single value up to the value 'alpha_u' sepecified in selpath() function)

Value

selvar:final selected variable

pval:pvalue of each selected variable (present if at least 1 var is selected)

devlist:deviance contributed by each selected variable (given the previous selected variables;present if at least 1 var is selected)

dist.mat:a list of deviance permutation distributions (including the distribution from the step from which no more variable is added)

`Subsample.w` *Subsample.w*

Description

This function is to subsample the data and perform LASSO (single time) on the selected samples

Usage

```
Subsample.w(data, fold, Alpha, prop, weights, family, Type)
```

Arguments

data	the dataframe should be arranged in the way such that columns are X1,X2,X3....,Xp, status. Where Xi's are variables and status is the outcome(for the logistic regression, the outcome is in terms of 0/1)
fold	fold used in lasso cross validation to select the tuning parameter
Type	should use 'class' for classification always
Alpha	1 for Lasso,0 for ridgeression
prop	percentage of samples left out for each subsamping
weights	=TRUE: if weighted version is desired; =FALSE, otherwise (binomial model);weights: =vector of weights of the same size as the sample size N: if weighted version is desired;=FALSE, otherwise (other generalized model)
family	the distribution family for the response variable.

Value

lambda: the tuning parameter that within 1 sd of the tuning parameter gives the lowest CV error

coef: a table shows the name of the selected variables by LASSO and its coefficients

table: there are a equal proportion of samples from each status left out and we use the model built on the selected

subsamples to predict those left out ones. Table contains two columns: column1 is the predicted value and column2 is the true class

error: misclassification error based on the above table

Beta: should be a vector of length p+1 and this is the beta coefficients from the LASSO model; Be aware of that the intercept is placed at the end of this vector

---

Subsample.w_cox *Subsample.w_cox*

Description

This function is to subsample the data and perform LASSO (single time) on the selected samples for cox proportional model

Usage

```
Subsample.w_cox(data, fold, Alpha, prop, weights)
```

Arguments

data	the dataframe should be arranged in the way such that columns are X1,X2,X3....,Xp, status. Where Xi's are variables and status is the outcome(for the logistic regression, the outcome is in terms of 0/1)
fold	fold used in lasso cross validation to select the tuning parameter
Alpha	1 for Lasso,0 for ridgeression
prop	percentage of samples left out for each sub-sampling
weights	= a vector of weights: if weighted version is desired, =FALSE, otherwise

Value

#lambda: the tuning parameter that within 1 sd of the tuning parameter gives the lowest CV error

coef: a table shows the name of the selected variables by LASSO and its coefficients

table: there are a equal proportion of samples from each status left out and we use the model built on the selected subsamples to predict those left out ones. Table contains two columns: column1 is the predicted value and column2 isthe true value of the outcome

error: misclassification error based on the above table

Beta: should be a vector of length p+1 and this is the beta coefficients from the LASSO model.

Subsample_B *Subsample_B*

Description

This function is to run sub-sampling B times

Usage

```
Subsample_B(B, data, fold, Alpha, prop, weights, ncores, family)
```

Arguments

B	the number of sub-samplings to run (e.g., B=1000)
data	the dataframe should be arranged in the way such that columns are X1,X2,X3....,Xp, status. Where Xi's are variables and status is the outcome(for the logistic regression, the outcome is in terms of 0/1)
fold	fold used in lasso cross validation to select the tuning parameter
Alpha	1 for Lasso,0 for ridgeression
prop	percentage of samples left out for each subsamping
family	The distribution family of the response variable
weights	In a binomial model, weights: =TRUE: if weighted version is desired; =FALSE, otherwise ; In other models,weights: =vector of weights of the same size as the sample size N: if weighted version is desired;=FALSE, otherwise (other generalized model)
ncores	the number of cores to use for parallel computation

Value

Class.Err: mis-classification error on the left out ones over B runs. A vector of length B.

Lambda: tuning parameters selected from B runs. It is a vector of length B

BETA: It is a matrix used to save the beta coefficients from all B runs #' @export

---

SURF *SURF*

Description

SuRF is a sparse variable selection method with uses a subsampling approach an LASSO to rank variables before applying forward selection using a permutation test. The function is able to give results at a range of significance levels simultaneously.

Usage

```
SURF(
  Xo,
  y,
  X = NULL,
  fold = 10,
  Alpha = 1,
  prop = 0.1,
  weights = FALSE,
  B = 1000,
  C = 200,
  ncores = 1,
  display.progress = TRUE,
  family = stats::binomial(link = "logit"),
  alpha_u = 0.1,
  alpha = 0.05
)
```

Arguments

Xo	- other type of predictor variables
y	- response variable, a vecotr for most families. For family="cox", y will should be a matrix of the response variable in column1 and censoring status in column 2.
X	- count data, need to be converted to proportion
fold	- number of folds for cross-validation in Lasso
Alpha	- Alpha parameter for elastic net
prop	- proportion of observations left out in subsampling

`weights` - use weighted regression: for unbalanced class sizes (bimomial family only) or weighted sample for other families;In a binomial model, weights: =TRUE: if weighted version is desired; =FALSE, otherwise ; In other models,weights: =vector of weights of the same size as the sample size N: if weighted version is desired;=FALSE, otherwise (other generalized model)

`B` - number of subsamples to take

`C` - number of permutations used to estimate null distribution

`display.progress` - whether SuRF should print a message on completion of each

`alpha_u` - the upper bound of significance level for the permutation test: alpha_u has to be in the range of (0,1). The large of this value, the longer the program will run;

`alpha` - the alpha value of interest (alpha >0 and must be <=alpha_u). It can be a single value or a vector.If missing, by default it is 0.05.

`ncores` whether SuRF should compute in parallel: 1 indicates NOT; anything greater will compute in parallel

`family` The distribution family of the response variable

Details

SuRF consists of two steps. In the first step, LASSO variable selection is applied to a large number of subsamples of the data set, to provide a list of selected variables for each subsample. This list is used to rank the variables, based on the number of subsamples in which each variable is selected, so that variables that are selected in more subsamples are ranked more highly. In the second step, this list is used as a basis for forward selection, with variables higher on the list tried first. If a highly-ranked variable is not selected, later variables are tried, and after each variable is selected, the variables not yet selected (even previously non-selected variables) are tried in order of the ranking from Step 1. The decision whether to include a variable is based on a permutation test for the deviance statistic.

Full details of the SuRF method are in the paper:

Lihui Liu, Hong Gu, Johan Van Limbergen, Toby Kenney (2020) SuRF: A new method for sparse variable selection, with application in microbiome data analysis Statistics in Medicine 40 897-919

doi: https://onlinelibrary.wiley.com/doi/10.1002/sim.8809

Value

Bmod: sub-sampling results

trdata: data frame including both X and y

ranklist: ranking table

modpath: variable selection path (along the alpha range)

selmod: model results at the selected alpha(s)

family: model family used

Examples

```
library(survival)
library(glmnet)
library(SuRF.vs)
N=100;p=200
nzc=p/3
X=matrix(rnorm(N*p),N,p)
beta=rnorm(nzc)
fx=X[,seq(nzc)]%*%beta/3
hx=exp(fx)
ty=rexp(N,hx)
tcens=rbinom(n=N,prob=.3,size=1)# censoring indicator (1 or 0)
Xo=NULL
B=20
Alpha=1
fold=5
ncores=1
prop=0.1
C=3
alpha_u=0.2
alpha=seq(0.01,0.1,len=20)

#binomial model
XX=X[,1:2]
f=1+XX%*%c(2,1.5)
p=exp(f)/(1+exp(f))
y=rbinom(100,1,p)
weights=FALSE
family=stats::binomial(link="logit")


surf_binary=SURF(Xo=X,y=y,fold=5,weights=weights,B=10,C=5,family=family,alpha_u=0.1,alpha=alpha)


#linear regression
y=1+XX%*%c(0.1,0.2)
family=stats::gaussian(link="identity")

surf_lm=SURF(Xo=X,y=y,fold=5,weights=weights,B=10,C=5,family=family,alpha_u=0.1,alpha=alpha)


#cox proportional model
y=cbind(time=ty,status=1-tcens)
weights=rep(1,100)
rseed=floor(runif(20,1,100))
weights[rseed]=2
family=list(family="cox")

surf_cox=SURF(Xo=X,y=y,fold=5,weights=weights,B=10,C=5,family=family,alpha_u=alpha_u,alpha=alpha)
```

update_dev *update_dev*

Description

This function is to derive the deviance distribution based on the permutation method This function is not to be used independently but will be called by the function selectnew()

Usage

```
update_dev(data, vslist, C, weights, ncores, family)
```

Arguments

data	the variable 'data' within seqcutoff()
vslist	a vector of selected variables
C	the number of permutation times
family	family=stats::gaussian(link="identity"));family=stats::binomial(link="logit");family=list(family="cox");
weights	In a binomial model, weights: =TRUE: if weighted version is desired; =FALSE, otherwise ; In other models,weights: =vector of weights of the same size as the sample size N: if weighted version is desired;=FALSE, otherwise (other generalized model)
ncores	the number of cores to use for parallel computation

Value

dev: a vector of deviance after C permutations (length OF this vector is C)

update_dev_cox *update_dev_cox*

Description

For COX proportional model ONLY. This function is to derive the deviance distribution based on the permutation method;This function is not to be used independently but will be called by the function selectnew()

Usage

```
update_dev_cox(data, vslist, C, ncores, weights)
```

Arguments

data	the variable 'data' within seqcutoff()
vslist	a vector of selected variables
C	the number of permutation times
weights	=TRUE: if weighted version is desired, =FALSE, otherwise (binomial model); weights: =vector of weights of the same size as the sample size N: if weighted version is desired, =FALSE, otherwise (other generalized model)
ncores	the number of cores to use for parallel computation

Value

dev: a vector of deviance after C permutations (length OF this vector is C)

Index