Package ‘SIS’

March 16, 2015

Version 0.7-5
Date 2015-03-15
Title Sure Independence Screening
Author Jianqing Fan, Yang Feng, Diego Franco Saldana, Richard Samworth, Yichao Wu
Maintainer Diego Franco Saldana <diego@stat.columbia.edu>
Depends R (>= 3.1.1), glmnet, ncvreg, survival

Description Variable selection techniques are essential tools for model selection and estimation in high-dimensional statistical models. Through this publicly available package, we provide a unified environment to carry out variable selection using iterative sure independence screening (SIS) and all of its variants in generalized linear models and the Cox proportional hazards model.

License GPL-2

R topics documented:

- leukemia.test .......................................................... 2
- leukemia.train .......................................................... 2
- predict.SIS ............................................................. 3
- print.SIS ............................................................... 5
- prostate.test .......................................................... 7
- prostate.train .......................................................... 8
- SIS ........................................................................... 8
- standardize ............................................................... 12
- tune.fit ................................................................. 13

Index 16
**leukemia.test**  
*Gene expression Leukemia testing dataset from Golub et al. (1999)*

**Description**

Gene expression testing data of 7129 genes from 34 patients with acute leukemias (20 in class Acute Lymphoblastic Leukemia and 14 in class Acute Myeloid Leukemia) from the microarray study of Golub et al. (1999).

**Usage**

`data(leukemia.test)`

**Format**

A data frame with 34 observations on 7129 variables.

**Source**

http://wwwprod.broadinstitute.org/cgi-bin/cancer/datasets.cgi

**References**


---

**leukemia.train**  
*Gene expression Leukemia training dataset from Golub et al. (1999)*

**Description**

Gene expression training data of 7129 genes from 38 patients with acute leukemias (27 in class Acute Lymphoblastic Leukemia and 11 in class Acute Myeloid Leukemia) from the microarray study of Golub et al. (1999).

**Usage**

`data(leukemia.train)`

**Format**

A data frame with 38 observations on 7129 variables.

**Source**

http://wwwprod.broadinstitute.org/cgi-bin/cancer/datasets.cgi
References


predict.SIS

Model prediction based on a fitted SIS object.

Description

Similar to the usual predict methods, this function returns predictions from a fitted "SIS" object.

Usage

## S3 method for class 'SIS'
predict(object, newx, s=object$path.index, type=c("response", "link", "class"), ...)

Arguments

object Fitted "SIS" model object.
newx Matrix of new values for \( x \) at which predictions are to be made, without the intercept term.
s Indices of the penalty parameter \( \lambda \) of the final fitted model by (I)SIS at which predictions are required. By default, only the index minimizing the criterion tune is returned.
type Type of prediction required. Type "response" gives the fitted values for "gaussian", fitted probabilities for "binomial", fitted mean for "poisson", and the fitted relative risk for "cox". Type "link" returns the linear predictors for "binomial", "poisson" and "cox" models; for "gaussian" models it is equivalent to type "response". Type "class" applies only to "binomial" models, and produces the class label corresponding to the maximum probability (0-1 labels).
... Not used. Other arguments to predict.

Value

The object returned depends on type.

Author(s)

Jianqing Fan, Yang Feng, Diego Franco Saldana, Richard Samworth, and Yichao Wu
References


Diego Franco Saldana and Yang Feng (2014) Sure Independence Screening in Ultrahigh Dimensional Statistical Models, manuscript.

See Also

*SIS*, *print.SIS*

Examples

```r
set.seed(0)
n = 400; p = 50; rho = 0.5
corrmat = diag(rep(1-rho, p)) + matrix(rho, p, p)
corrmat[,4] = sqrt(rho)
corrmat[4, ] = sqrt(rho)
corrmat[4,4] = 1
corrmat[,5] = 0
corrmat[5, ] = 0
corrmat[5,5] = 1
cholmat = chol(corrmat)
x = matrix(rnorm(n*p, mean=0, sd=1), n, p)
x = x%*%cholmat
testX = matrix(rnorm(10*p, mean=0, sd=1), nrow=10, ncol=p)

# gaussian response
set.seed(1)
b = c(4,4,4,-6*sqrt(2),4/3)
y=x[, 1:5]*%*%b + rnorm(n)
model1=SIS(x, y, family="gaussian", tune="bic", varISIS="aggr", seed=11)
predict(model1, testX, type="response")
predict(model1, testX, s=1:10, type="response")

# binary response
set.seed(2)
leta = x[, 1:5]*%*%b; fprob = exp(leta)/(1+exp(leta))
y = rbinom(n, 1, fprob)
```
print.SIS

Print a SIS object

Description
Print a summary of the selected predictors and its estimated coefficients.

Usage
## S3 method for class 'SIS'
print(x, digits = max(3, getOption("digits") - 3), ...)
Arguments

x Fitted "SIS" model object.
digits Significant digits in printout.
... Additional print arguments.

Details

The call that produced object x is printed, followed by the indices of the selected predictors (i.e., x$ix) and its estimated coefficients (i.e., x$coef.est) for the final model picked up by (I)SIS.

Value

The fitted regression object of type "cv.ncvreg", "ncvreg", "cv.glmnet", or "glmnet" and its corresponding path index where the minimum criterion in tune occurs are silently returned.

Author(s)

Jianqing Fan, Yang Feng, Diego Franco Saldana, Richard Samworth, and Yichao Wu

References


Diego Franco Saldana and Yang Feng (2014) Sure Independence Screening in Ultrahigh Dimensional Statistical Models, manuscript.

See Also

*SIS, predict.SIS*

Examples

```r
set.seed(0)
n = 400; p = 50; rho = 0.5
corrmat = diag(rep(1-rho, p)) + matrix(rho, p, p)
corrmat[,4] = sqrt(rho)
corrmat[4,] = sqrt(rho)
corrmat[4,4] = 1
corrmat[,5] = 0
```
```r
corrmat[5, ] = 0
corrmat[5,5] = 1
cholmat = chol(corrmat)
x = matrix(rnorm(n*p, mean=0, sd=1), n, p)
x = x%*%cholmat

# gaussian response
set.seed(1)
b = c(4,4,4,-6*sqrt(2),4/3)
y=x[, 1:5]%*%b + rnorm(n)
model1=sls(x, y, family="gaussian", tune="bic", varISO="aggr", seed=11)

print(model1, digits=3)
print(model1, digits=7)
```

---

**prostate.test**  
*Gene expression Prostate Cancer testing dataset from Singh et al. (2002)*

**Description**

Gene expression testing data of 12600 genes from 25 patients with prostate tumors and 9 normal specimens from the microarray study of Singh et al. (2002).

**Usage**

```r
data(prostate.test)
```

**Format**

A data frame with 34 observations on 12600 variables.

**Source**

http://wwwprod.broadinstitute.org/cgi-bin/cancer/datasets.cgi

**References**

prostate.train  

*Gene expression Prostate Cancer training dataset from Singh et al. (2002)*

**Description**

Gene expression training data of 12600 genes from 52 patients with prostate tumors and 50 normal specimens from the microarray study of Singh et al. (2002).

**Usage**

```r
data(prostate.train)
```

**Format**

A data frame with 102 observations on 12600 variables.

**Source**

http://wwwprod.broadinstitute.org/cgi-bin/cancer/datasets.cgi

**References**


---

**SIS**

*(Iterative) Sure Independence Screening ((I)SIS) and Fitting in Generalized Linear Models and Cox's Proportional Hazards Models*

**Description**

This function first implements the Iterative Sure Independence Screening for different variants of (I)SIS, and then fits the final regression model using the R packages *ncvreg* and *glmnet* for the SCAD/MCP/LASSO regularized loglikelihood for the variables picked by (I)SIS.

**Usage**

```r
SIS(x, y, family = c("gaussian","binomial","poisson","cox"), 
penalty=c("SCAD","MCP","lasso"), concavity.parameter = 
switch(penalty, SCAD=3.7, 3), tune = c("cv","aic","bic","ebic"), 
nfolds = 10, type.measure = c("deviance","class","auc","mse", 
"mae"), gamma.ebic = 1, nsis = NULL, iter = TRUE, iter.max = 
ifelse(greedy==FALSE,10,floor(nrow(x)/log(nrow(x)))))
varISIS = 
c("vanilla","aggr","cons"), perm = FALSE, q = 1, greedy = FALSE, 
greedy.size = 1, seed = 0, standardize = TRUE)
```
Arguments

\( x \)  
The design matrix, of dimensions \( n \times p \), without an intercept. Each row is an observation vector. SIS standardizes the data and includes an intercept by default.

\( y \)  
The response vector of dimension \( n \times 1 \). Quantitative for \( \text{family} = \text{"gaussian"}, \) non-negative counts for \( \text{family} = \text{"poisson"}, \) binary (0-1) for \( \text{family} = \text{"binomial"}. \) For \( \text{family} = \text{"cox"}, y \) should be an object of class \text{Surv}, as provided by the function \text{Surv}() in the package \text{survival}.

\text{family}  
Response type (see above).

\text{penalty}  
The penalty to be applied in the regularized likelihood subproblems. "SCAD" (the default), "MCP", or "lasso" are provided.

\text{concavity parameter}  
The tuning parameter used to adjust the concavity of the SCAD/MCP penalty. Default is 3.7 for SCAD and 3 for MCP.

\text{tune}  
Method for tuning the regularization parameter of the penalized likelihood subproblems and of the final model selected by (I)SIS. Options include \text{tune} = "cv", \text{tune} = "aic", \text{tune} = "bic", and \text{tune} = "ebic".

\text{nfolds}  
Number of folds used in cross-validation. The default is 10.

\text{type.measure}  
Loss to use for cross-validation. Currently five options, not all available for all models. The default is \text{type.measure} = "deviance", which uses squared-error for gaussian models (also equivalent to \text{type.measure} = "mse" in this case), deviance for logistic and poisson regression, and partial-likelihood for the Cox model. Both \text{type.measure} = "class" and \text{type.measure} = "auc" apply only to logistic regression and give misclassification error and area under the ROC curve, respectively. \text{type.measure} = "mse" or \text{type.measure} = "mae" (mean absolute error) can be used by all models except the "cox"; they measure the deviation from the fitted mean to the response. For \text{penalty} = "SCAD" and \text{penalty} = "MCP", only \text{type.measure} = "deviance" is available.

\text{gamma.ebic}  
Specifies the parameter in the Extended BIC criterion penalizing the size of the corresponding model space. The default is \text{gamma.ebic} = 1. See references at the end for details.

\text{nsis}  
Number of predictors recruited by (I)SIS.

\text{iter}  
Specifies whether to perform iterative SIS. The default is \text{iter} = TRUE.

\text{iter.max}  
Maximum number of iterations for (I)SIS and its variants.

\text{varISIS}  
Specifies whether to perform any of the two ISIS variants based on randomly splitting the sample into two groups. The variant \text{varISIS} = "aggr" is an aggressive variable screening procedure, while \text{varISIS} = "cons" is a more conservative approach. The default is \text{varISIS} = "vanilla", which performs the traditional vanilla version of ISIS. See references at the end for details.

\text{perm}  
Specifies whether to impose a data-driven threshold in the size of the active sets calculated during the ISIS procedures. The threshold is calculated by first decoupling the predictors \( x \), and response \( y \), through a random permutation \( \pi \) of \((1, \ldots, n)\) to form a null model. For this newly permuted data, marginal regression coefficients for each predictor are recalculated. As the marginal regression coefficients of the original data should be larger than most recalculated coefficients in the null model, the data-driven threshold is given by the \( q \)th quantile of
the null coefficients. This data-driven threshold only allows a $1 - q$ proportion of inactive variables to enter the model when $x_i$ and $y_i$ are not related (in the null model). The default is here is perm=FALSE. See references at the end for details.

$q$
Quantile for calculating the data-driven threshold in the permutation-based ISIS. The default is $q=1$ (i.e., the maximum absolute value of the permuted estimates).

greedy
Specifies whether to run the greedy modification of the permutation-based ISIS. The default is greedy=FALSE.

greedy.size
Maximum size of the active sets in the greedy modification of the permutation-based ISIS. The default is greedy.size=1.

seed
Random seed used for sample splitting, random permutation, and cross-validation sampling of training and test sets.

standardize
Logical flag for x variable standardization, prior to performing (iterative) variable screening. The resulting coefficients are always returned on the original scale. Default is standardize=TRUE. If variables are in the same units already, you might not wish to standardize.

Value

Returns an object with

ix
The vector of indices selected by (I)SIS.

coef.est
The vector of coefficients of the final model selected by (I)SIS.

fit
A fitted object of type ncvreg, cv.ncvreg, glmnet, or cv.glmnet for the final model selected by the (I)SIS procedure. If tune="cv", the returned fitted object is of type cv.ncvreg if penalty="SCAD" or penalty="MCP"; otherwise, the returned fitted object is of type cv.glmnet. For the remaining options of tune, the returned object is of type glmnet if penalty="lasso", and ncvreg otherwise.

path.index
The index along the solution path of fit for which the criterion specified in tune is minimized.

Author(s)

Jianqing Fan, Yang Feng, Diego Franco Saldana, Richard Samworth, and Yichao Wu

References


Diego Franco Saldana and Yang Feng (2014) Sure Independence Screening in Ultrahigh Dimensional Statistical Models, manuscript.


See Also

`predict.SIS`, `print.SIS`

Examples

```r
set.seed(0)
n = 400; p = 50; rho = 0.5
corrmat = diag(rep(1-rho, p)) + matrix(rho, p, p)
corrmat[,1] = sqrt(rho)
corrmat[4,] = sqrt(rho)
corrmat[4,4] = 1
corrmat[,5] = 0
corrmat[5,] = 0
corrmat[5,5] = 1
cholmat = chol(corrmat)
x = matrix(rnorm(n*p, mean=0, sd=1), n, p)
x = x%*%cholmat

# gaussian response
set.seed(1)
b = c(4,4,4,-6*sqrt(2),4/3)
y=x[, 1:5]%*%b + rnorm(n)
model11=SIS(x, y, family="gaussian", tune="bic")
model12=SIS(x, y, family="gaussian", tune="bic", varISIS="aggr", seed=11)
model11$ix
model12$ix

# binary response
set.seed(2)
leta = x[, 1:5]%*%b; fprob = exp(leta)/(1+exp(leta))
y = rbinom(n, 1, fprob)
model21=SIS(x, y, family="binomial", tune="bic")
model22=SIS(x, y, family="binomial", tune="bic", varISIS="aggr", seed=21)
model21$ix
model22$ix

# poisson response
set.seed(3)
b = c(0.6,0.6,0.6,-0.9*sqrt(2))
myrates = exp(x[, 1:4]%*%b)
y = rpois(n, myrates)
```
Standardization of High-Dimensional Design Matrices

**Description**

Standardizes the columns of a high-dimensional design matrix to mean zero and unit Euclidean norm.

**Usage**

`standardize(X)`

**Arguments**

- `X`  
  A design matrix to be standardized.

**Details**

Performs a location and scale transform to the columns of the original design matrix, so that the resulting design matrix with \( p \)-dimensional observations \( \{ x_i : i = 1, ..., n \} \) of the form \( x_i = (x_{i1}, x_{i2}, ..., x_{ip}) \) satisfies \( \sum_{i=1}^{n} x_{ij} = 0 \) and \( \sum_{i=1}^{n} x_{ij}^2 = 1 \) for \( j = 1, ..., p \).

**Value**

A design matrix with standardized predictors or columns.
tune.fit

Author(s)
Jianqing Fan, Yang Feng, Diego Franco Saldana, Richard Samworth, and Yichao Wu

References
Diego Franco Saldana and Yang Feng (2014) Sure Independence Screening in Ultrahigh Dimensional Statistical Models, manuscript.

Examples

set.seed(0)
n = 400; p = 50; rho = 0.5
corrmat = diag(rep(1-rho, p)) + matrix(rho, p, p)
corrmat[,4] = sqrt(rho)
corrmat[4,] = sqrt(rho)
corrmat[4,4] = 1
corrmat[,5] = 0
corrmat[5,] = 0
corrmat[5,5] = 1
cholmat = chol(corrmat)
x = matrix(rnorm(n*p, mean=15, sd=9), n, p)
x = x%*%cholmat

x.standard = standardize(x)

Using the \texttt{glmnet} and \texttt{ncvreg} packages, fits a Generalized Linear Model or Cox Proportional Hazards Model using various methods for choosing the regularization parameter $\lambda$

Description
This function fits a generalized linear model or a Cox proportional hazards model via penalized maximum likelihood, with available penalties as indicated in the \texttt{glmnet} and \texttt{ncvreg} packages. Instead of providing the whole regularization solution path, the function returns the solution at a unique value of $\lambda$, the one optimizing the criterion specified in tune.

Usage

tune.fit(x, y, family = c("gaussian","binomial","poisson","cox"),
penalty=c("SCAD","MCP","lasso"), concavity.parameter =
switch(penalty, SCAD=3.7, 3), tune = c("cv","aic","bic","ebic"),
nfolds = 10, type.measure = c("deviance","class","auc","mse",
"mae"), gamma.ebic = 1)
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. Quantitative for family="gaussian",
non-negative counts for family="poisson", binary (0-1) for family="binomial".
For family="cox", y should be an object of class Surv, as provided by the func-
tion Surv() in the package survival.

family  Response type (see above).

penalty  The penalty to be applied in the regularized likelihood subproblems. "SCAD"
(the default), "MCP", or "lasso" are provided.

concavity parameter  The tuning parameter used to adjust the concavity of the SCAD/MCP penalty.
Default is 3.7 for SCAD and 3 for MCP.

tune  Method for selecting the regularization parameter along the solution path of
the penalized likelihood problem. Options to provide a final model include
tune="cv", tune="aic", tune="bic", and tune="ebic". See references at
the end for details.

nfolds  Number of folds used in cross-validation. The default is 10.

type.measure  Loss to use for cross-validation. Currently five options, not all available for all
models. The default is type.measure="deviance", which uses squared-error
for gaussian models (also equivalent to type.measure="mse" in this case), de-
viance for logistic and poisson regression, and partial-likelihood for the Cox
model. Both type.measure="class" and type.measure="auc" apply only
to logistic regression and give misclassification error and area under the ROC
curve, respectively. type.measure="mse" or type.measure="mae" (mean ab-
solute error) can be used by all models except the "cox"; they measure the deviation
from the fitted mean to the response. For penalty="SCAD" and penalty="MCP",
only type.measure="deviance" is available.

gamma.ebic  Specifies the parameter in the Extended BIC criterion penalizing the size of the
corresponding model space. The default is gamma.ebic=1. See references at the
end for details.

Value

Returns an object with

ix  The vector of indices of the nonzero coefficients selected by the maximum pe-
nalized likelihood procedure with tune as the method for choosing the regular-
ization parameter.

a0  The intercept of the final model selected by tune.

fit  The vector of coefficients of the final model selected by tune.

Author(s)

Jianqing Fan, Yang Feng, Diego Franco Saldana, Richard Samworth, and Yichao Wu
References


Examples

```r
tune.fit
```
```r
set.seed(8)
data("leukemia.train", package = "SIS")
y.train = leukemia.train[,dim(leukemia.train)[2]]
x.train = as.matrix(leukemia.train[,dim(leukemia.train)[2]])
x.train = standardize(x.train)
model = tune.fit(x.train[,1:3500], y.train, family="binomial", tune="bic")
model$ix
model$a0
model$fit
```
Index

*Topic datasets
  leukemia.test, 2
  leukemia.train, 2
  prostate.test, 7
  prostate.train, 8

*Topic models
  predict.SIS, 3
  print.SIS, 5
  SIS, 8
  standardize, 12
  tune.fit, 13

leukemia.test, 2
leukemia.train, 2

predict.SIS, 3, 6, 11
print.SIS, 4, 5, 11
prostate.test, 7
prostate.train, 8

SIS, 4, 6, 8
standardize, 12

tune.fit, 13