Package 'VSURF'

May 28, 2013

Type Package

Title Variable Selection Using Random Forests

Version 0.5

Date 2013-05-28

Author Robin Genuer, Jean-Michel Poggi and Christine Tuleau-Malot

Maintainer Robin Genuer <Robin.Genuer@isped.u-bordeaux2.fr>

Description Three steps variable selection procedure based on random forests. Initially developed to handle high dimensional data (for which number of variables largely exceeds number of observations),the package is very versatile and can treat most dimensions of data, for regression and supervised classification problems.
First step is dedicated to eliminate irrelevant variables from the dataset.
Second step aims to select all variables related to the response for interpretation purpose.
Third step refines the selection by eliminating redundancy in the set of variables selected by the second step, for prediction purpose.

License GPL (>= 2)

Depends randomForest, rpart

R topics documented:

VSURF-package *Variable Selection Using Random Forests*

Description

Three steps variable selection procedure based on random forests. Initially developped to handle high dimensional data (for which number of variables largely exceeds number of observations), the package is very versatile and can treat most dimensions of data, for regression and supervised classification problems. First step is dedicated to eliminate irrelevant variables from the dataset. Second step aims to select all variables related to the response for interpretation prupose. Third step refines the selection by eliminating redundancy in the set of variables selected by the second step, for prediction prupose.

Details

Package:	VSURF
Type:	Package
Version:	1.0
Date:	2012-07-19
License:	GPL (>= 2)

The most important function is the function VSURF.

Author(s)

Robin Genuer, Jean-Michel Poggi and Christine Tuleau-Malot

Maintainer: <Robin.Genuer@isped.u-bordeaux2.fr>

References

Genuer, R. and Poggi, J.M. and Tuleau-Malot, C. (2010), *Variable selection using random forests*, Pattern Recognition Letters 31(14), 2225-2236

See Also

VSURF

VSURF *Variable Selection Using Random Forests*

Description

Three steps variable selection procedure based on random forests for supervised classification problems (for regression problems, see...) First step ("thresholding step") is dedicated to eliminate irrelevant variables from the dataset. Second step ("interpretation step") aims to select all variables related to the response for interpretation prupose. Third step ("prediction step") refines the selection by eliminating redundancy in the set of variables selected by the second step, for prediction prupose.

Usage

```
VSURF(x, y, ntree=500, nfor.thres=50, nmin=1,
      nfor.interp=25, nsd=1, nfor.pred=25, nmj=1,
      mtry=if (!is.factor(y)) max(floor(ncol(x)/3), 1)
           else floor(sqrt(ncol(x)))
      )
```

Arguments

x	A data frame or a matrix of predictors, the columns represent the variables.
y	A response vector (must be a factor for classification problems and numeric for regression ones).
ntree	Number of trees in each forests grown. Standard parameter of randomForest.
mtry	Number of variables randomly sampled as candidates at each split. Standard parameter of randomForest.
nfor.thres	Number of forests grown for "thresholding step" (first of the three steps).
nmin	Number of times the "minimum value" is multiplied to set threshold value.
nfor.interp	Number of forests grown for "intepretation step" (second of the three steps).
nsd	Number of times the standard deviation of the minimum value of err.interp is multiplied.
nfor.pred	Number of forests grown for "prediction step" (last of the three steps).
nmj	Number of times the mean jump is multiplied.

Details

- First step ("thresholding step"): first, nfor.thres random forests are computed using the function randomForest with arguments importance=TRUE. Then variables are sorted according to their mean variable importance (VI), in decreasing order. This order is kept all along the procedure. Next, a threshold is computed: min.thres, the minimum predicted value of a pruned CART tree fitted to the curve of the standard deviations of VI. Finally, the actual "thresholding step" is performed: only variables with a mean VI larger than nmin * min.thres are kept.
- Second step ("intepretation step"): the variables selected by the first step are considered. nfor.interp embedded random forests models are grown, starting with the random forest build with only the most important variable and ending with all variables selected in the first step. Then, err.min the minimum mean out-of-bag (OOB) error of these models and its

associated standard deviation sd.min are computed. Finally, the smallest model (and hence its corresponding variables) having a mean OOB error less than err.min + nsd * sd.min is selected.

- Third step ("prediction step"): The starting point is the same than in the second step. However, now the variables are added to the model in a stepwise manner. mean.jump, the mean jump value is calculated using variables that have been left out by the second step, and is set as the mean absolute difference between mean OOB errors of one model and its first following model. Hence a variable is included in the model if the mean OOB error decrease is larger than nmj * mean.jump.

Value

An object of class VSURF, which is a list with the following components:

varselect.thres
: A vector of indexes of variables selected after "thresholding step", sorted according to their mean VI, in decreasing order.

imp.varselect.thres
: A vector of importances of the varselect.thres variables.

min.thres
: The minimum predicted value of a pruned CART tree fitted to the curve of the standard deviations of VI.

ord.imp
: A list containing the order of all variables mean importance. $x contains the mean importances sorted in decreasing order. $ix contains indexes of the variables.

ord.sd
: A vector of standard deviations of all variables importance. The order is given by ord.imp.

mean.perf
: The mean OOB error rate, obtained by a random forests build on all variables.

varselect.interp
: A vector of indexes of variables selected after "interpretation step".

err.interp
: A vector of the mean OOB error rates of the embedded random forests models build during the "interpretation step".

sd.min
: The standard deviation of OOB error rates associated to the random forests model attaining the minimum mean OOB error rate during the "interpretation step".

varselect.pred
: A vector of indexes of variables selected after "prediction step".

err.pred
: A vector of the mean OOB error rates of the random forests models build during the "prediction step".

mean.jump
: The mean jump value computed during the "prediction step".

Author(s)

Robin Genuer, Jean-Michel Poggi and Christine Tuleau-Malot

References

Genuer, R. and Poggi, J.M. and Tuleau-Malot, C. (2010), Variable selection using random forests, Pattern Recognition Letters 31(14), 2225-2236

Examples

```
data(iris)
iris.vsurf <- VSURF(x=iris[,1:4], y=iris[,5], ntree=100, nfor.thres=20,
                    nfor.interp=10, nfor.pred=10)
```

VSURF.interp *Interpretation step of VSURF for supervised classification problems*

Description

Interpretation step aims to select all variables related to the response for interpretation prupose. This is the second step of the VSURF function for supervised classification problems. It is designed to be executed after the thresholding step VSURF.thres.

Usage

```
VSURF.interp(x, y, vars, nfor.interp = 25, nsd = 1)
```

Arguments

x	A data frame or a matrix of predictors, the columns represent the variables.
y	A response vector (must be a factor for classification problems and numeric for regression ones).
vars	A vector of variable indices. Typically, indices of variables selected by thresholding step (see value varselect.thres of VSURF.thres function).
nfor.interp	Number of forests grown.
nsd	Number of times the standard deviation of the minimum value of err.interp is multiplied. See details below.

Details

nfor.interp embedded random forests models are grown, starting with the random forest build with only the most important variable and ending with all variables. Then, err.min the minimum mean out-of-bag (OOB) error of these models and its associated standard deviation sd.min are computed. Finally, the smallest model (and hence its corresponding variables) having a mean OOB error less than err.min + nsd * sd.min is selected.

Value

An object of class VSURF.interp, which is a list with the following components:

varselect.interp	A vector of indices of selected variables.
err.interp	A vector of the mean OOB error rates of the embedded random forests models.
sd.min	The standard deviation of OOB error rates associated to the random forests model attaining the minimum mean OOB error rate.

Author(s)

Robin Genuer, Jean-Michel Poggi and Christine Tuleau-Malot

References

Genuer, R. and Poggi, J.M. and Tuleau-Malot, C. (2010), *Variable selection using random forests*, Pattern Recognition Letters 31(14), 2225-2236

See Also

VSURF

Examples

```
data(iris)
iris.thres <- VSURF.thres(x=iris[,1:4], y=iris[,5], ntree=100, nfor.thres=20)
iris.interp <- VSURF.interp(x=iris[,1:4], y=iris[,5], vars=iris.thres$varselect.thres,
                            nfor.interp=10)
```

VSURF.pred *Prediction step of VSURF for supervised classification problems*

Description

Prediction step refines the selection of intepretation step VSURF.interp by eliminating redundancy in the set of variables selected, for prediction prupose.This is the third step of the VSURF function for supervised classification problems.

Usage

```
VSURF.pred(x, y, err.interp, varselect.interp, nfor.pred = 25, nmj = 1)
```

Arguments

x	A data frame or a matrix of predictors, the columns represent the variables.
y	A response vector (must be a factor for classification problems and numeric for regression ones).
err.interp	A vector of the mean OOB error rates of the embedded random forests models build during interpretation step (value err.interp of function VSURF.interp).
varselect.interp	A vector of indices of variables selected after interpretation step.
nfor.pred	Number of forests grown.
nmj	Number of times the mean jump is multiplied. See details below.

Details

nfor.pred embedded random forests models are grown, starting with the random forest build with only the most important variable. Variables are added to the model in a stepwise manner. The mean jump value mean.jump is calculated using variables that have been left out by interpretation step, and is set as the mean absolute difference between mean OOB errors of one model and its first following model. Hence a variable is included in the model if the mean OOB error decrease is larger than nmj * mean.jump.

Value

An object of class VSURF.pred, which is a list with the following components:

varselect.pred	A vector of indices of variables selected after "prediction step".
err.pred	A vector of the mean OOB error rates of the random forests models build during the "prediction step".
mean.jump	The mean jump value computed during the "prediction step".

Author(s)

Robin Genuer, Jean-Michel Poggi and Christine Tuleau-Malot

References

Genuer, R. and Poggi, J.M. and Tuleau-Malot, C. (2010), *Variable selection using random forests*, Pattern Recognition Letters 31(14), 2225-2236

See Also

VSURF

Examples

```
data(iris)
iris.thres <- VSURF.thres(x=iris[,1:4], y=iris[,5], ntree=100, nfor.thres=20)
iris.interp <- VSURF.interp(x=iris[,1:4], y=iris[,5], vars=iris.thres$varselect.thres,
                            nfor.interp=10)
iris.pred <- VSURF.pred(x=iris[,1:4], y=iris[,5], err.interp=iris.interp$err.interp,
varselect.interp=iris.interp$varselect.interp, nfor.pred=10)
```

VSURF.thres *Thresholding step of VSURF for supervised classification problems*

Description

Thresholding step is dedicated to roughly eliminate irrelevant variables a the dataset. This is the first step of the VSURF function for supervised classification problems. For refined variable selection, see the VSURF other steps: VSURF.interp and VSURF.pred.

Usage

```
VSURF.thres(x, y, ntree=500, nfor.thres=50, nmin=1,
mtry=if (!is.factor(y)) max(floor(ncol(x)/3), 1) else floor(sqrt(ncol(x))) )
```

Arguments

x	A data frame or a matrix of predictors, the columns represent the variables.
y	A response vector (must be a factor for classification problems and numeric for regression ones).
ntree	Number of trees in each forest grown. Standard randomForest parameter.
nfor.thres	Number of forests grown.
nmin	Number of times the "minimum value" is multiplied to set threshold value. See details below.
mtry	Number of variables randomly sampled as candidates at each split. Standard randomForest parameter.

Details

First, nfor.thres random forests are computed using the function randomForest with arguments importance=TRUE. Then variables are sorted according to their mean variable importance (VI), in decreasing order. This order is kept all along the procedure. Next, a threshold is computed: min.thres, the minimum predicted value of a pruned CART tree fitted to the curve of the standard deviations of VI. Finally, the actual thresholding is performed: only variables with a mean VI larger than nmin * min.thres are kept.

Value

An object of class VSURF.thres, which is a list with the following components:

varselect.thres	A vector of indices of selected variables, sorted according to their mean VI, in decreasing order.
imp.varselect.thres	A vector of importances of the varselect.thres variables.
min.thres	The minimum predicted value of a pruned CART tree fitted to the curve of the standard deviations of VI.
ord.imp	A list containing the order of all variables mean importance. $x contains the mean importances in decreasing order. $ix contains indices of the variables.
ord.sd	A vector of standard deviations of all variables importances. The order is given by ord.imp.
mean.perf	The mean OOB error rate, obtained by a random forests build with all variables.

Author(s)

Robin Genuer, Jean-Michel Poggi and Christine Tuleau-Malot

References

Genuer, R. and Poggi, J.M. and Tuleau-Malot, C. (2010), *Variable selection using random forests*, Pattern Recognition Letters 31(14), 2225-2236

See Also

VSURF

Examples

```
data(iris)
iris.thres <- VSURF.thres(x=iris[,1:4], y=iris[,5], ntree=100, nfor.thres=20)
```

Index