Package 'VSURF'

July 26, 2013

Type Package

Title Variable Selection Using Random Forests

Version 0.6

Date 2013-07-26

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 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.

Details

Package:	VSURF
Type:	Package
Version:	0.6
Date:	2013-07-26
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`

plot *Plot of VSURF results*

Description

This function plots 4 graphs illustrating VSURF results.

Usage

```
## S3 method for class 'VSURF'
plot(x, ...)
```

Arguments

x	An object of class VSURF, which is the result of the VSURF function.
...	...

Details

The 2 graphs of the top row correspond to the "thresholding step". The top left graph plots the mean variable importance in decreasing order (black curve). The red horizontal line represent the value of the threshold. The top right graph plots the standard deviation of variable importance with variables ordered according to their mean variable importance in decreasing order (black curve). The green line represents the predictions given by a CART tree fitted to the black curve (the standard deviations). Finally, the dotted horizontal red line represents the minimum value of the CART predictions, which actually is the value of the threshold.

The bottom left graph corresponds to the "interpretation step". It plots the mean OOB error rate of embedded random forests models (from the one with only one variable as predictor, to the one with all variables kept after the "thresholding step"). The vertical red line indicates the retained model.

The bottom right graph corresponds to the "predicton step". It plots the mean OOB error rate of embedded random forests models (the difference, here, being that variables are added to the model in a step-wise manner). The retained model is the final one.

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, summary.VSURF

Examples

```
## Not run:
data(iris)
iris.vsurf <- VSURF(x=iris[,1:4], y=iris[,5], ntree=100, nfor.thres=20,
                    nfor.interp=10, nfor.pred=10)
plot(iris.vsurf)

# A more interesting example with toys data (see ?toys)
# (less than 1 min to execute)
data(toys)
```

```
toys.vsurf <- VSURF(x=toys$x, y=toys$y)
plot(toys.vsurf)
## End(Not run)
```

---

summary *Summary of VSURF results*

Description

This function display a summary of VSURF results

Usage

```
## S3 method for class 'VSURF'
summary(object, ...)
```

Arguments

object	An object of class VSURF, which is the result of the VSURF function.
...	...

Details

This function prints the total computation time of VSURF. It also gives the number of selected variables at each step of VSURF.

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, plot.VSURF

Examples

```
## Not run:
data(iris)
iris.vsurf <- VSURF(x=iris[,1:4], y=iris[,5], ntree=100, nfor.thres=20,
                    nfor.interp=10, nfor.pred=10)
summary(iris.vsurf)

# A more interesting example with toys data (see ?toys)
# (less than 1 min to execute)
```

```
data(toys)
toys.vsurf <- VSURF(x=toys$x, y=toys$y)
summary(toys.vsurf)
## End(Not run)
```

`toys` *A simulated dataset called toys data*

Description

`toys` is a simple simulated dataset of a binary classification problem, introduced by Weston et.al..

Usage

```
data(toys)
```

Format

The format is a list of 2 component:

$x: A data-frame containing input variables: with 100 obs. of 200 variables

$y: Outpu variable: a factor with 2 levels "-1" and "1"

Details

It is an equiprobable two class problem, Y belongs to -1,1, with six true variables, the others being some noise. The simulation model is defined through the conditional distribution of the Xi for Y=y:

with probability 0.7, X^j ~ N(yj,1) for j=1,2,3 and X^j ~ N(0,1) for j=4,5,6.

with probability 0.3, X^j ~ N(0,1) for j=1,2,3 and X^j ~ N(y(j-3),1) for j=4,5,6.

the other variables are noise, X^j ~ N(0,1) for j=7,. . . ,p.

After simulation, the obtained variables are finally standardized.

Source

Weston, J., Elisseff, A., Schoelkopf, B., Tipping, M. (2003), *Use of the zero norm with linear models and Kernel methods*, J. Machine Learn. Res. 3, 1439-1461

Examples

```
data(toys)
system.time(toys.rf <- randomForest(x=toys$x, y=toys$y))
toys.rf

## Not run:
# VSURF applied for toys data:
# (less than 1 min to execute)
data(toys)
toys.vsurf <- VSURF(x=toys$x, y=toys$y)
toys.vsurf
## End(Not run)
```

VSURF *Variable Selection Using Random Forests*

Description

Three steps variable selection procedure based on random forests for supervised classification and regression problems. 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,
      mtry=if (!is.factor(y)) max(floor(ncol(x)/3), 1)
           else floor(sqrt(ncol(x))),
      nfor.thres=50, nmin=1, nfor.interp=25, nsd=1, 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).
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.

`num.varselect.thres`
: Number of variables selected by "thresholding step".

`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`
: Mean OOB error rate, obtained by a random forests build on all variables.

`pred.pruned.treee`
: Predictions of the CART tree fitted to the curve of the standard deviations of VI.

`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".

`num.varselect.interp`
: Number of variables selected by "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".
num.varselect.pred	Number of variables selected by "prediction step".
nmin	Number of times the "minimum value" is multiplied to set threshold value.
nsd	Number of times the standard deviation of the minimum value of err.interp is multiplied.
nmj	Number of times the mean jump is multiplied.
comput.time	Overall computation time

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

plot.VSURF, summary.VSURF, VSURF.thres, VSURF.interp, VSURF.pred

Examples

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

## Not run:
# A more interesting example with toys data (see ?toys)
# (less than 1 min to execute)
data(toys)
toys.vsurf <- VSURF(x=toys$x, y=toys$y)
toys.vsurf
## End(Not run)
```

VSURF.interp *Interpretation step of VSURF*

Description

Interpretation step aims to select all variables related to the response for interpretation prupose. This is the second step of the VSURF function. 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 rate 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

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

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, VSURF.interp.tune

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.interp

## Not run:
# A more interesting example with toys data (see ?toys)
# (less than 1 min to execute)
data(toys)
toys.thres <- VSURF.thres(x=toys$x, y=toys$y)
toys.interp <- VSURF.interp(x=toys$x, y=toys$y, vars=toys.thres$varselect.thres)
toys.interp
## End(Not run)
```

VSURF.interp.tune *Tuning of the interpretation step of VSURF*

Description

This function allows to tune the "interpretation step" of VSURF, without rerunning all computations.

Usage

```
VSURF.interp.tune(res.interp, nsd = 1)
```

Arguments

res.interp	An object of class VSURF.interp, which is the result of the VSURF.interp function.
nsd	Number of times the standard deviation of the minimum value of err.interp is multiplied. See details below.

Details

In VSURF.interp function, the smallest model (and hence its corresponding variables) having a mean OOB error rate less than err.min + nsd * sd.min is selected. The function VSURF.interp.tune allows to change the value of nsd (which multiply the standard deviation of the minimum OOB error rate sd.min), without rerunning all computations. To get a larger model than default, choose a value of nsd less than 1, and to get a smaller one, choose a value larger than 1.

Value

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.

`num.varselect.interp`
: The number of selected variables.

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

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`, `VSURF.interp`

Examples

```
## Not run:
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.interp
iris.interp.tuned <- VSURF.interp.tune(res.interp=iris.interp, nsd=10)
iris.interp.tuned
## End(Not run)
```

VSURF.pred *Prediction step of VSURF*

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.

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

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".
num.varselect.pred	The number of selected 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)
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)
iris.pred

## Not run:
# A more interesting example with toys data (see ?toys)
# (less than 1 min to execute)
data(toys)
toys.thres <- VSURF.thres(x=toys$x, y=toys$y)
toys.interp <- VSURF.interp(x=toys$x, y=toys$y, vars=toys.thres$varselect.thres)
toys.pred <- VSURF.pred(x=toys$x, y=toys$y, err.interp=toys.interp$err.interp,
                        varselect.interp=toys.interp$varselect.interp)
toys.pred
## End(Not run)
```

`VSURF.thres` *Thresholding step of VSURF*

Description

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

Usage

```
VSURF.thres(x, y, ntree=500,
            mtry=if (!is.factor(y)) max(floor(ncol(x)/3), 1)
                 else floor(sqrt(ncol(x))),
            nfor.thres=50, nmin=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).
`ntree`	Number of trees in each forest grown. Standard `randomForest` parameter.
`mtry`	Number of variables randomly sampled as candidates at each split. 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.

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

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.

`num.varselect.thres`
: The number of selected variables.

`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.

`pred.pruned.treee`
: The predictions of the CART tree fitted to the curve of the standard deviations of VI.

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`, `VSURF.thres.tune`

Examples

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

## Not run:
# A more interesting example with toys data (see ?toys)
# (less than 1 min to execute)
data(toys)
toys.thres <- VSURF.thres(x=toys$x, y=toys$y)
toys.thres
## End(Not run)
```

VSURF.thres.tune *Tuning of the thresholding step of VSURF*

Description

This function allows to tune the "thresholding step" of VSURF, without rerunning all computations.

Usage

```
VSURF.thres.tune(res.thres, nmin = 1)
```

Arguments

res.thres	An object of class VSURF.thres, which is the result of the VSURF.thres function.
nmin	Number of times the "minimum value" is multiplied to set threshold value. See details below.

Details

In VSURF.thres function, the actual threshold is performed like this: only variables with a mean VI larger than nmin * min.thres are kept. The function VSURF.thres.tune allows you to change the value of nmin (which multiply the estimated threshold value min.thres), without rerunning all computations. To get a softer threshold than default, choose a value of nmin less than 1, and to get a harder one, choose a value larger than 1.

Value

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.

`num.varselect.thres` The number of selected variables.

`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.

`pred.pruned.treee` The predictions of the CART tree fitted to the curve of the standard deviations of VI.

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`, `VSURF.thres`

Examples

```
## Not run:
data(iris)
iris.thres <- VSURF.thres(x=iris[,1:4], y=iris[,5], ntree=100, nfor.thres=20)
iris.thres.tuned <- VSURF.thres.tune(res.thres=iris.thres, nmin=10)
iris.thres.tuned
## End(Not run)
```

Index