Package 'sivs'

October 14, 2022

Type Package

Title Stable Iterative Variable Selection

Version 0.2.5

Date 2021-07-20

Imports doParallel, parallel, foreach, glmnet, pROC, varhandle, utils

Suggests knitr, rmarkdown, markdown

Description An iterative feature selection method (manuscript submitted) that internally utilizes various Machine Learning methods that have embedded feature reduction in order to shrink down the feature space into a small and yet robust set.

License GPL-3

URL https://github.com/mmahmoudian/sivs,

https://doi.org/10.1093/bioinformatics/btab501

BugReports https://github.com/mmahmoudian/sivs/issues

Encoding UTF-8

RoxygenNote 7.1.1

VignetteBuilder knitr

NeedsCompilation no

Author Mehrad Mahmoudian [aut, cre] (<https://orcid.org/0000-0001-7650-1862>),
Mikko Venäläinen [aut, rev] (<https://orcid.org/0000-0003-1777-4259>),
Riku Klèn [aut, ths] (<https://orcid.org/0000-0002-0982-8360>),
Laura Elo [aut, ths, fnd] (<https://orcid.org/0000-0001-5648-4532>)

Maintainer Mehrad Mahmoudian <mehrad.mahmoudian@utu.fi>

Repository CRAN

Date/Publication 2021-07-20 15:50:02 UTC

R topics documented:

`coef.sivs` *Extract Coefficients from sivs object*

Description

A function to extract the coefficients of "iterative.res" step or any part of "rfe" such as "sivs_objectrfebaseline" from a sivs object.

Usage

```
## S3 method for class 'sivs'
coef(object, step = "iterative.res", ...)
```

Arguments

`object`	An object of class "sivs"
`step`	A a character string of length 1. It should either specify the step ("iterative.res" or "rfe"), or step$subsetp (e.g "rfe$baseline").
`...`	potential further arguments (required for Method/Generic reasons).

Value

The function returns a data.frame that has features as rows and different runs as columns, with the addition of the first column which contains the feature name.

Examples

```
## Not run:
# getting the coefficients of features for the baseline runs in rfe
coef(object = sivs_object, step = "rfe$baseline")

## End(Not run)

## WORKING EXAMPLE
## Note that this example does not logically make sense as iris data has only
## 4 columns and there is no need for SIVS to take care of feature selection
## therefore this example is only here for testing purposes.
```

```
tmp <- subset(x = iris, subset = Species != "setosa")

tmp <- varhandle::unfactor(tmp)

sivs_obj <- sivs(x = tmp[, c("Sepal.Length", "Sepal.Width",
                             "Petal.Length", "Petal.Width")],
                 y = factor(tmp$Species),
                 family = "binomial",
                 verbose = "detailed",
                 progressbar = FALSE,
                 nfolds = 3,
                 parallel.cores = FALSE,
                 iter.count = 20)

coef(sivs_obj)
```

`is.sivs` *Validate sivs Object*

Description

A function to validate if the given object is truely from sivs function

Usage

```
is.sivs(object)
```

Arguments

`object`	Idealy the object that is produced by the sivs function.

Value

This function will return TRUE if it detects the function is truely a sivs object, otherwise it will return FALSE.

`plot.sivs` *A plotting function for sivs object*

Description

A function to plot the object of the sivs function.

Usage

```
## S3 method for class 'sivs'
plot(
  x,
  type = c("frequency", "coef", "rfe"),
  suggestion_strictness = c(0.01, 0.05),
  ...
)
```

Arguments

x	The object that is produced by the sivs function.
type	Which plot do you want to have. Acceptable values are "frequency", "coef", and "rfe".
suggestion_strictness	The strictness value that indicates how much the thresholds should be strict. For more details visit help page of ‘suggest()‘ function. If you want ru suppress the suggestion on the plot, set this to NULL.
...	The other argument you might want to pass to each plot.

Details

The rfe plot ignores the existance of intercept since intercept cannot be removed in the recursive feature elimination step. This is the reason that the number of features with vimp greater than 0 are differnt from the presented "Number of Features Left" in the rfe plot by 1.

Value

Does not return anything. This function is only used for plotting and is not meant to return any value.

Examples

```
## Not run:
# to see all plots
layout(mat = matrix(c(1,2,3,3), nrow = 2, byrow = TRUE))
plot(x = sivs_object)
layout(1)

# to plot only the Recursive Feature Elimination (rfe) results
plot(x = sivs_object, type = "rfe")

# suppress suggestion on rfe plot
plot(x = sivs_object, type = "rfe", suggestion_strictness = NULL)

## End(Not run)

## WORKING EXAMPLE
## Note that this example does not logically make sense as iris data has only
## 4 columns and there is no need for SIVS to take care of feature selection
```

```
## therefore this example is only here for testing purposes.

tmp <- subset(x = iris, subset = Species != "setosa")

tmp <- varhandle::unfactor(tmp)

sivs_obj <- sivs(x = tmp[, c("Sepal.Length", "Sepal.Width",
                             "Petal.Length", "Petal.Width")],
                 y = factor(tmp$Species),
                 family = "binomial",
                 verbose = "detailed",
                 progressbar = FALSE,
                 nfolds = 3,
                 parallel.cores = FALSE,
                 iter.count = 20)
plot(sivs_obj, type = "frequency")
plot(sivs_obj, type = "coef")
plot(sivs_obj, type = "rfe")
```

`sivs` *Stable Iterative Variable Selection*

Description

The name is an acronym for Stable Iterative Variable Selection. This function will iteratively run a machine learning method that can incorporate a shrinkage method using multiple random seeds in order to find the smallest set of features that can robustly be predictive.

Usage

```
sivs(
  x,
  y,
  test.ratio = 1/3,
  method = "glmnet",
  iter.count = 100,
  nfolds = 10,
  sample.grouping = NULL,
  parallel.cores = "grace",
  progressbar = TRUE,
  verbose = "general",
  return.fits = FALSE,
  return.roc = FALSE,
  return.sessionInfo = TRUE,
  lib.paths = .libPaths(),
  debug.mode = FALSE,
  ...
)
```

Arguments

x	The input data. Each row should represent a sample and each column should represent a feature.
y	Response variable. It should be of class factor for classification and of class Surv for survival.
test.ratio	How much of the data should be cut and used for testing
method	The internal machine learning method to be used
iter.count	How many iterations should the function go through
nfolds	How many folds should the training cross-validations have
sample.grouping	A character, numeric or factor vector to specify how the samples should be grouped/bundled together in the cross-validation binning. If set to NULL the grouping will be skipped. Samples with the same value will always be kept together in the same bins in cross-validation. This is especially useful when having multiple samples from the same individual. Default is NULL.
parallel.cores	How many cores should be used in the iterative process. The value should be the number of threads in numeric form, or any of these values: "max", "grace", FALSE, NULL. If set to "max", all cores will be used and in large datasets you might face your computer struggling and ultimately errors. If set to "grace", one core will be left out so that it can be used by other processes in the machine. If set to NULL of FALSE, the code will run sequentially and without the parallel backend.
progressbar	Logical. If the progressbar should be shown. Default is TRUE.
verbose	Character. How detailed the progress should be reported. The value should be a character vector of length 1. "detailed" will report every single step. "general" will report only main steps. "none" or FALSE will suppress any reporting.
return.fits	Logical. Whether the fit object for each iterative run should be returned. Having the fits in the final object would significantly increase the final object size. Default is FALSE.
return.roc	Logical. Whether the ROC object for each iterative run should be returned. Having the fits in the final object would significantly increase the final object size. Default is FALSE.
return.sessionInfo	Logical. Whether the utils::sessionInfo() be included in the final object. This is useful for reproducibility purposes. Default is TRUE.
lib.paths	A character vector that contains the paths that the dependency libraries are in it. REMEMBER to set this if you are using packrat.
debug.mode	Whether or not the debug mode should be enabled.
...	Other parameters to be passed to the training method. For example the value of alpha in glmnet.

Value

An object with S3 class "sivs". run.info$call: The call that produced this object run.info$sessionInfo: The object produced by utils::sessionInfo()

Examples

```
## Not run:
# considering that you have your data object as `DATA` where you have rows
# as samples and columns as features, and the response value as a vector
# named `RESP`:

# simple defult run
sivs_object <- sivs(x = DATA, y = RESP)

# simple run with using only 3 CPU cores
sivs_object <- sivs(x = DATA, y = RESP, parallel.cores = 3)


# get the variable importance values
sivs_object$vimp

# get the condision that the sivs was ran in
sivs_object$run.info$call
sivs_object$run.info$sessionInfo

## End(Not run)

## WORKING EXAMPLE
## Note that this example does not logically make sense as iris data has only
## 4 columns and there is no need for SIVS to take care of feature selection
## therefore this example is only here for testing purposes.

tmp <- subset(x = iris, subset = Species != "setosa")

tmp <- varhandle::unfactor(tmp)

sivs_obj <- sivs(x = tmp[, c("Sepal.Length", "Sepal.Width",
                             "Petal.Length", "Petal.Width")],
                 y = factor(tmp$Species),
                 family = "binomial",
                 verbose = "detailed",
                 progressbar = FALSE,
                 nfolds = 3,
                 parallel.cores = FALSE,
                 iter.count = 20)
```

`str.sivs` *Structure of sivs object*

Description

This function shows the structure of a an object of either class "sivs" or "list" and shows the internal structure of the object in human-readable format. sivs object is a complex S3 object and it might be

a deterrent to users to get to know if better. This function is aiming to facilitate the experience of user.

Usage

```
## S3 method for class 'sivs'
str(
  object,
  max_depth = 2,
  max_leaves = 2,
  max_width = options("width")$width,
  ...
)
```

Arguments

object	An object of class "sivs" or "list".
max_depth	A numerical value of length 1 indicating how many layers the function should dive into.
max_leaves	A numericl vector of length 1 indicating how many of the objects of each list should be shown.
max_width	A numerical vector of length 1 indicating the maximum width of the terminal that should be used by the function. If there are any lines larger than this value, they will be truncated. Default is the terminal size that is retuned by R.
...	potential further arguments (required for Method/Generic reasons).

Value

The function uses ‘cat‘ to output general structure of sivs object in human readable format in a tree-like structure.

Examples

```
## WORKING EXAMPLE
## Note that this example does not logically make sense as iris data has only
## 4 columns and there is no need for SIVS to take care of feature selection
## therefore this example is only here for testing purposes.

tmp <- subset(x = iris, subset = Species != "setosa")

tmp <- varhandle::unfactor(tmp)

sivs_obj <- sivs(x = tmp[, c("Sepal.Length", "Sepal.Width",
                             "Petal.Length", "Petal.Width")],
                 y = factor(tmp$Species),
                 family = "binomial",
                 verbose = "detailed",
                 progressbar = FALSE,
                 nfolds = 3,
```

```
                  parallel.cores = FALSE,
                  iter.count = 20)

str(sivs_obj)
```

---

suggest *Cutoff Suggestion for sivs Object*

Description

A function to suggest the user a set of features based on sivs object and the provided strictness

Usage

```
suggest(object, strictness = 0.01, plot = FALSE)
```

Arguments

object	The object that is produced by the sivs function.
strictness	A numerical vector of length 1 showing how strict the suggestion should be, ranging from 0 to 1 where 0 is less strict and 1 is the most strict. Default value is 0.01. For more information, check the Details section.
plot	A logical vector of length 1 indicating whether the suggestion should also be plotted in the "rfe" plot. The same plot can be generated via plot.sivs() function when the suggestion_strictness is set according to the strictness argument of this function.

Details

This function tries to narrow down the list of VIMP features in sivs object into a smaller feature list based on provided strictness coefficient. This function practically defines a threshold for AUCs in the rfe (Recursive Feature Elimination) step of sivs. Any run with any set of features that are above the AUC threshold will be eliminated and all the features that were contributing into having an AUC lower than the threshold are returned. The cutoff is defined as: ((1 - strictness) * (max(median_AUROCs) - min(median_AUROCs))) + min(median_AUROCs) where median_AUROCs is the median of AUROCs for each run in rfe step of sivs. Note that this function is supposed to give the feature space based on the cutoff and hence the intercept (if exists in the VIMP) will be excluded from the output.

Value

A character vector that contains the names of suggested features based on the defined strictness.

Examples

```
## Not run:
# Defult use
suggest(sivs_object)

# get the suggested features and also plot it with strictness of 0.01
suggest(object = sivs_object, strictness = 0.01, plot = TRUE)

## End(Not run)

## WORKING EXAMPLE
## Note that this example does not logically make sense as iris data has only
## 4 columns and there is no need for SIVS to take care of feature selection
## therefore this example is only here for testing purposes.

tmp <- subset(x = iris, subset = Species != "setosa")

tmp <- varhandle::unfactor(tmp)

sivs_obj <- sivs(x = tmp[, c("Sepal.Length", "Sepal.Width",
                             "Petal.Length", "Petal.Width")],
                 y = factor(tmp$Species),
                 family = "binomial",
                 verbose = "detailed",
                 progressbar = FALSE,
                 nfolds = 3,
                 parallel.cores = FALSE,
                 iter.count = 20)

suggest(sivs_obj)
```

Index