Package ‘nestfs’

May 9, 2022

Type Package
Title Cross-Validated (Nested) Forward Selection
Version 1.0.2
Date 2022-05-04
Description Implementation of forward selection based on cross-validated linear and logistic regression.
License GPL-2 | file LICENSE
URL https://github.com/mcol/nestfs
BugReports https://github.com/mcol/nestfs/issues
Imports dgof, parallel, pROC (>= 1.9), methods, stats, utils
Suggests testthat (>= 2.0.0)
Encoding UTF-8
RoxygenNote 7.1.2
NeedsCompilation no
Author Marco Colombo [aut, cre] (https://orcid.org/0000-0001-6672-0623),
Felix Agakov [ctb]
Maintainer Marco Colombo <mar.colombo13@gmail.com>
Repository CRAN
Date/Publication 2022-05-09 12:10:05 UTC

R topics documented:

   nestfs-package .................................................. 2
   create.folds .................................................... 3
   diabetes .......................................................... 3
   fs ................................................................. 4
   nested.fs ......................................................... 7
   nested.glm ........................................................ 8
   nested.performance ............................................... 9
   summary.fs ....................................................... 10
   summary.nestfs .................................................. 11
Description

This package provides an implementation of forward selection based on linear and logistic regression which adopts cross-validation as a core component of the selection procedure.

Details

The engine of the package is \texttt{fs()}, whose aim is to select a set of variables out of those available in the dataset. The selection of variables can be done according to two main different criteria: by paired-test p-value or by largest decrease in validation log-likelihood. A combined criteria is also available.

The role of \texttt{nested.fs()} is to allow the evaluation of the selection method by providing an unbiased estimate of the performance of the selected variables on withdrawn data.

Forward selection is an inherently slow approach, as for each variable a model needs to be fitted. In our implementation, this issue is further aggravated by the fact that an inner cross-validation happens at each iteration, with the aim of guiding the selection towards variables that have better generalization properties.

The code is parallelized over the inner folds, thanks to the \texttt{parallel} package. User time therefore depends on the number of available cores, but there is no advantage in using more cores than inner folds. The number of cores assigned to computations must be registered before starting by setting the "mc.cores" option.

The main advantage of forward selection is that it provides an immediately interpretable model, and the panel of variables obtained is in some sense the least redundant one, particularly if the number of variables to choose from is not too large (in our experience, up to about 30-40 variables).

However, when the number of variables is much larger than that, forward selection, besides being unbearably slow, may be more subject to overfitting, which is in the nature of its greedy-like design. These undesirable effects can be somewhat remedied by applying some filtering (see the \texttt{num.filter} argument to \texttt{fs()}, thus reducing the number or variables entering the selection phase.

Author(s)

Marco Colombo <mar.colombo13@gmail.com>

See Also

Useful links:

- \url{https://github.com/mcol/nestfs}
- Report bugs at \url{https://github.com/mcol/nestfs/issues}
create.folds

Cross-validation folds

Description
Create a list of indices corresponding to cross-validation folds.

Usage
create.folds(num.folds, num.rows, seed = NULL)

Arguments
- **num.folds**: Number of folds to be created.
- **num.rows**: Number of observations in the dataset.
- **seed**: Seed of the random number generator. If NULL, the folds generated will be different at each invocation; for reproducibility of results, it is recommended to set this to a specific value.

Value
A list of length num.folds containing the indices of the observations to be withdrawn for validation in each fold.

Note
Note that the number of observations withdrawn in each fold may not be exactly the same if num.folds is not an integer divisor of num.rows.

Examples
all.folds <- create.folds(50, 307, 0)

diabetes

Diabetes data with interaction terms

Description
The dataset consists of observations on 442 individuals for which a quantitative measure of diabetes progression is recorded in variable $Y$. Predictors include 10 baseline measurements, 45 interactions and 9 quadratic terms, for a total of 64 variables for each individual. All predictors have been standardized by subtracting the mean and then dividing by the standard deviation.
Source


The original dataset is available from https://web.stanford.edu/~hastie/Papers/LARS/data64.txt

Examples

data(diabetes, package="nestfs")

fs  Cross-validated forward selection

Description

Run forward selection starting from a baseline model. As it uses all observations in the input data frame, it is not possible to produce unbiased estimates of the predictive performance of the panel selected (use nested.fs() for that purpose).

Usage

fs(
  formula,
  data,
  family,
  choose.from = NULL,
  test = c("t", "wilcoxon"),
  num.inner.folds = 30,
  max.iters = 10,
  min.llk.diff = 2,
  max.pval = 0.5,
  sel.crit = c("paired.test", "total.loglik", "both"),
  num.filter = 0,
  filter.ignore = NULL,
  seed = 50,
  verbose = TRUE
)

forward.selection(x, y, init.model, family, ...)

Arguments

formula An object of class formula (or one that can be coerced to that class) that describes the baseline model to be fitted.

data Data frame or matrix containing outcome variable and predictors.
family
Type of model fitted: either `gaussian()` for linear regression or `binomial()` for logistic regression. This can be specified also as a function name (`gaussian`) or as a string ("gaussian").

choose.from
Indices or variable names over which the selection should be performed. If `NULL` (default), all variables in `x` that are not in `init.model` are considered.

test
Type of statistical paired test to use (ignored if `sel.crit="total.loglik"`).

num.inner.folds
Number of folds in the inner cross-validation. It must be at least 5 (default: 30).

max.iter
Maximum number of iterations (default: 10).

min.llk.diff
Minimum improvement in log-likelihood required before selection is terminated (default: 2).

max.pval
Interrupt the selection when the best achievable p-value exceeds this threshold (default: 0.5).

sel.crit
Selection criterion: "paired.test" chooses the variable with smallest p-value using the paired test specified by `test` (see Details), as long as this is smaller than `max.pval`; "total.loglik" picks the variable that gives the largest increase in log-likelihood; "both" attempts to combine both previous criteria, choosing the variable that produces the largest increase in log-likelihood only among the best 5 variables ranked according to the paired-test p-value.

num.filter
Number of variables to be retained by the univariate association filter (see Details), which can only be enabled if `family=binomial()`. Variables listed in `init.model` are never filtered. If set to 0 (default), the filter is disabled.

filter.ignore
Vector of variable names that should not be pruned by the univariate association filter so that they are always allowed to be selected (ignored if `num.filter=0`).

seed
Seed of the random number generator for the inner folds.

verbose
Whether the variable chosen at each iteration should be printed out (default: `TRUE`).

x
Dataframe of predictors: this should include all variables in the initial set and the variables that are allowed to enter the selected panel.

y
Outcome variable. If `family=binomial`, it can only contain two classes of values that can be coerced to 0-1.

init.model
Either a formula or a vector of names of the initial set of variables that define the model from which the forward selection should start.

Details
At each iteration, this function runs cross-validation to choose which variable enters the final panel by fitting the current model augmented by each remaining variable considered one at a time.

By default variables are selected according to the `paired.test` criterion. At each iteration, the sampling distribution of differences in validation log-likelihood obtained across all inner cross-validation folds of the models with and without each additional variable are tested against the null hypothesis of zero mean (with the alternative hypothesis being that the model with the additional variable is better). The test is paired according to the inner folds. Although the training folds are
not independent, the p-value from this test approximates the probability that including the marker will not decrease the validation log-likelihood (approximate false discovery rate).

In the case of a binary outcome when very large number of predictors is available, it may be convenient to apply a univariate association filter. If num.filter is set to a positive value, then all available predictors (excluding those whose name is matched by filter.ignore) are tested for univariate association with the outcome, and only the first num.filter enter the selection phase, while the others are filtered out. This is done on the training part of all inner folds. Filtering can enhance the performance of forward selection when the number of available variables exceeds about 30-40.

forward.selection provides the legacy interface used up to version 0.9.2. It is considered discontinued, and in the future it will be deprecated and eventually removed.

**Value**

An object of class fs containing the following fields:

- `fs`: A data frame containing the forward selection summary.
- `init`: The set of variables used in the initial model.
- `panel`: Names of variables selected (in order).
- `init.model`: Right-hand side of the formula corresponding to the initial model.
- `final.model`: Right-hand side of the formula corresponding to the final model after forward selection.
- `family`: Type of model fitted.
- `params`: List of parameters used.
- `iter1`: Summary statistics for all variables at the first iteration.
- `all.iter`: Validation log-likelihoods for all inner folds at all iterations.

**See Also**

`nested.fs()` and `summary.fs()`.

**Examples**

```r
data(diabetes)
fs.res <- fs(Y ~ age + sex, data=diabetes, family=gaussian(),
             choose.from=1:10, num.inner.folds=5, max.iters=3)
summary(fs.res)
```
nested.fs

**Nested cross-validated forward selection**

**Description**

Run nested forward selection starting from a set of variables or a model.

**Usage**

```r
nested.fs(formula, data, family, folds, ...)
```

```r
nested.forward.selection(x, y, init.model, family, folds, ...)
```

**Arguments**

- `formula`: An object of class `formula` (or one that can be coerced to that class) that describes the baseline model to be fitted.
- `data`: Data frame or matrix containing outcome variable and predictors.
- `family`: Type of model fitted: either `gaussian()` for linear regression or `binomial()` for logistic regression. This can be specified also as a function name (`gaussian`) or as a string ("gaussian").
- `folds`: List of cross-validation folds, where each element contains the indices of the observations to be withdrawn in that fold.
- `...`: Arguments to `fs()`.
- `x`: Dataframe of predictors: this should include all variables in the initial set and the variables that are allowed to enter the selected panel.
- `y`: Outcome variable. If `family=binomial`, it can only contain two classes of values that can be coerced to 0-1.
- `init.model`: Either a formula or a vector of names of the initial set of variables that define the model from which the forward selection should start.

**Details**

This function allows to obtain an unbiased estimate of the performance of the selected panels on withdrawn data by running forward selection on a predetermined set of folds.

`nested.forward.selection` provides the legacy interface used up to version 0.9.2. It is considered discontinued, and in the future it will be deprecated and eventually removed.

**Value**

An object of class `nestfs` of length equal to `length(folds)`, where each element is an object of class `fs` containing the following additional fields:

- `fit`: Predicted values for the withdrawn observations.
- `obs`: Observed values for the withdrawn observations.
- `test.idx`: Indices of the the withdrawn observations for this fold.
- `model`: Summary of the model built using the selected panel.
See Also

fs(), summary.nestfs() and nested.performance().

Examples

data(diabetes)
folds <- create.folds(2, nrow(diabetes), seed=1)
nestfs.res <- nested.fs(Y ~ age + sex, diabetes, gaussian(), folds,
                       choose.from=1:10, num.inner.folds=5, max.iters=3)
summary(nestfs.res)

nested.glm

Cross-validated generalized linear models

Description

Run linear or logistic regression on a set of cross-validation folds. This can be used to establish a baseline model, often built only on the initial set of covariates.

Usage

nested.glm(formula, data, family, folds, store.glm = FALSE)

Arguments

formula An object of class formula (or one that can be coerced to that class) that describes the baseline model to be fitted.
data Data frame or matrix containing outcome variable and predictors.
family Type of model fitted: either gaussian() for linear regression or binomial() for logistic regression. This can be specified also as a function name (gaussian) or as a string ("gaussian").
folds List of cross-validation folds, where each element contains the indices of the observations to be withdrawn in that fold.
store.glm Whether the object produced by glm should be stored (default: FALSE).

Value

An object of class nestglm of length equal to length(folds), where each entry contains the following fields:

summary Summary of the coefficients of the model fitted on the training observations.
family Type of model fitted.
fit Predicted values for the withdrawn observations.
nested.performance

obs Observed values for the withdrawn observations.
test.1lk Test log-likelihood.
test.idx Indices of the the withdrawn observations for this fold.
regr Object created by glm (only if store.glm=TRUE).

See Also

nested.performance().

Examples

data(diabetes)
folds <- create.folds(10, nrow(diabetes), seed=1)
res <- nested.glm(Y ~ age + sex + bmi + map, diabetes, gaussian(), folds)

Description

Compute an unbiased estimate of the performance of a given model or forward selected panel using the results obtained on the cross-validation folds.

Usage

nested.performance(x)

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

Arguments

x An object of class nestfs or nestglm.
digits Number of significant figures to print.
... Further arguments passed to or from other methods. These are currently ignored.

Value

An object of class nestperf containing the following fields:

observed Vector of observed values from all folds.
predicted Vector of predicted values from all folds.
performance A performance measure: the area under the curve (AUC) if family="binomial", or the correlation coefficient if family="gaussian".


See Also

`nested.fs()` and `nested.glm()`.

---

### summary.fs

**Description**

Report summary statistics from a single run of forward selection.

**Usage**

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

```r
## S3 method for class 'fs'
print(x, ...)
```

**Arguments**

- `object, x` An object of class `fs`.
- `...` Further arguments passed to or from other methods. These are currently ignored.

**Value**

A data frame with the following columns:

- `vars` Variables in the initial model followed by variables selected.
- `fdr` False discovery rate, corresponding to the paired test p-values computed when the variable was selected.
- `llks` Validation log-likelihoods.
- `difs` Differences in validation log-likelihoods.
- `iter` Iteration when the variable was selected.

**Note**

A function of name "getfullname" to match variable names to full names is searched on the current workspace, and if found full names are included in the summary data frame.
**summary.nestfs**

---

**Results summary for nested forward selection**

**Description**

Report summary statistics from a run of nested forward selection across the outer folds.

**Usage**

```r
## S3 method for class 'nestfs'
summary(object, iter1 = FALSE, ...)

## S3 method for class 'nestfs'
print(x, ...)
```

**Arguments**

- `object, x`: An object of class `nestfs`.
- `iter1`: Whether the summary should be over all variables at the first iteration: this can be interpreted as a cross-validated univariate test for association.
- `...`: Further arguments passed to or from other methods. These are currently ignored.

**Value**

A data frame with the following columns:

- `vars`: Variables selected.
- `percent`: Percentage of folds in which the variable was selected.
- `coef`: Median coefficient for the variable.
- `coefIQR`: Inter-quartile range for the variable coefficient.
- `rank`: Median iteration in which the variable was selected.
- `rankIQR`: Inter-quartile range for rank of the variable.
- `diffLogLik`: Median difference in log-likelihoods.
- `diffLogLikIQR`: Inter-quartile range for the difference in log-likelihoods.

**Note**

A function of name "getfullname" to match variable names to full names is searched on the current workspace, and if found full names are included in the summary data frame.
Index

* datasets
  diabetes, 3
* multivariate
  fs, 4
  nested.fs, 7
  nested.glm, 8
create.folds, 3
diabetes, 3
forward.selection(fs), 4
fs, 4
fs(), 2, 7, 8
nested.forward.selection(nested.fs), 7
nested.fs, 7
nested.fs(), 2, 4, 6, 10
nested.glm, 8
nested.glm(), 10
nested.performance, 9
nested.performance(), 8, 9
nestfs(nestfs-package), 2
nestfs-package, 2
print.fs(summary.fs), 10
print.nestfs(summary.nestfs), 11
print.nestperf(nested.performance), 9
summary.fs, 10
summary.fs(), 6
summary.nestfs, 11
summary.nestfs(), 8