Package: varImp (via r-universe)

September 5, 2024

Title RF Variable Importance for Arbitrary Measures

Version 0.4

Description Computes the random forest variable importance (VIMP) for
the conditional inference random forest (cforest) of the
'party' package. Includes a function (varImp) that computes the
VIMP for arbitrary measures from the 'measures' package. For
calculating the VIMP regarding the measures accuracy and AUC
two extra functions exist (varImpACC and varImpAUC).

License GPL-3

Encoding UTF-8

Depends R (>= 3.0.0), measures, party, stats

LazyData true

Date 2020-05-15

RoxygenNote 7.0.2

Suggests testthat, ranger

Repository https://philipppro.r-universe.dev

RemoteUrl https://github.com/philipppro/varimp

RemoteRef HEAD

RemoteSha c7b544c80e2332cebb20f25e22a68dcf5efb63aa

Contents

---

varImp	*varImp*

Description

Computes the variable importance for arbitrary measures from the 'measures' package.

Usage

```
varImp(
  object,
  mincriterion = 0,
  conditional = FALSE,
  threshold = 0.2,
  nperm = 1,
  OOB = TRUE,
  pre1.0_0 = conditional,
  measure = "multiclass.Brier",
  ...
)
```

Arguments

object	An object as returned by cforest.
mincriterion	The value of the test statistic or 1 - p-value that must be exceeded in order to include a split in the computation of the importance. The default mincriterion = 0 guarantees that all splits are included.
conditional	a logical determining whether unconditional or conditional computation of the importance is performed.
threshold	The threshold value for (1 - p-value) of the association between the variable of interest and a covariate, which must be exceeded inorder to include the covariate in the conditioning scheme for the variable of interest (only relevant if conditional = TRUE). A threshold value of zero includes all covariates.
nperm	The number of permutations performed.
OOB	A logical determining whether the importance is computed from the out-of-bag sample or the learning sample (not suggested).
pre1.0_0	Prior to party version 1.0-0, the actual data values were permuted according to the original permutation importance suggested by Breiman (2001). Now the assignments to child nodes of splits in the variable of interest are permuted as described by Hapfelmeier et al. (2012), which allows for missing values in the explanatory variables and is more efficient wrt memory consumption and computing time. This method does not apply to conditional variable importances.
measure	The name of the measure of the 'measures' package that should be used for the variable importance calculation.
...	Further arguments (like positive or negativ class) that are needed by the measure.

Details

Many measures have not been tested for the usefulness of random forests variable importance. Use at your own risk.

Value

Vector with computed permutation importance for each variable.

Examples

```
# multiclass case
data(iris)
iris.cf = cforest(Species ~ ., data = iris, control = cforest_unbiased(mtry = 2, ntree = 50))
set.seed(123)
vimp = varImp(object = iris.cf, measure = "multiclass.Brier")
vimp
```

---

varImpACC   *varImpACC*

Description

Computes the variable importance regarding the accuracy (ACC).

Usage

```
varImpACC(
  object,
  mincriterion = 0,
  conditional = FALSE,
  threshold = 0.2,
  nperm = 1,
  OOB = TRUE,
  pre1.0_0 = conditional
)
```

Arguments

object	An object as returned by cforest.
mincriterion	The value of the test statistic or 1 - p-value that must be exceeded in order to include a split in the computation of the importance. The default mincriterion = 0 guarantees that all splits are included.
conditional	The value of the test statistic or 1 - p-value that must be exceeded in order to include a split in the computation of the importance. The default mincriterion = 0 guarantees that all splits are included.

threshold	The threshold value for (1 - p-value) of the association between the variable of interest and a covariate, which must be exceeded inorder to include the covariate in the conditioning scheme for the variable of interest (only relevant if conditional = TRUE). A threshold value of zero includes all covariates.
nperm	The number of permutations performed.
OOB	A logical determining whether the importance is computed from the out-of-bag sample or the learning sample (not suggested).
pre1.0_0	Prior to party version 1.0-0, the actual data values were permuted according to the original permutation importance suggested by Breiman (2001). Now the assignments to child nodes of splits in the variable of interest are permuted as described by Hapfelmeier et al. (2012), which allows for missing values in the explanatory variables and is more efficient wrt memory consumption and computing time. This method does not apply to conditional variable importances.

Value

Vector with computed permutation importance for each variable

Examples

```
data(iris)
iris2 = iris
iris2$Species = factor(iris$Species == "versicolor")
iris.cf = cforest(Species ~ ., data = iris2,control = cforest_unbiased(mtry = 2, ntree = 50))
set.seed(123)
a = varImpACC(object = iris.cf)
```

---

varImpAUC	*varImpAUC*

Description

Computes the variable importance regarding the AUC. Bindings are not taken into account in the AUC definition as they did not provide as good results as the version without bindings in the paper of Janitza et. al (2013) (see References section).

Usage

```
varImpAUC(
  object,
  mincriterion = 0,
  conditional = FALSE,
  threshold = 0.2,
  nperm = 1,
  OOB = TRUE,
  pre1.0_0 = conditional
)
```

Arguments

`object`	An object as returned by cforest.
`mincriterion`	The value of the test statistic or 1 - p-value that must be exceeded in order to include a split in the computation of the importance. The default mincriterion = 0 guarantees that all splits are included.
`conditional`	The value of the test statistic or 1 - p-value that must be exceeded in order to include a split in the computation of the importance. The default mincriterion = 0 guarantees that all splits are included.
`threshold`	The threshold value for (1 - p-value) of the association between the variable of interest and a covariate, which must be exceeded inorder to include the co-variate in the conditioning scheme for the variable of interest (only relevant if conditional = TRUE). A threshold value of zero includes all covariates.
`nperm`	The number of permutations performed.
`OOB`	A logical determining whether the importance is computed from the out-of-bag sample or the learning sample (not suggested).
`pre1.0_0`	Prior to party version 1.0-0, the actual data values were permuted according to the original permutation importance suggested by Breiman (2001). Now the assignments to child nodes of splits in the variable of interest are permuted as described by Hapfelmeier et al. (2012), which allows for missing values in the explanatory variables and is more efficient wrt memory consumption and computing time. This method does not apply to conditional variable importances.

Details

For using the original AUC definition and multiclass AUC you can use the varImp function and specify the particular measure.

Value

Vector with computed permutation importance for each variable

References

https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-14-119

Examples

```
# multiclass case
data(iris)
iris2 = iris
iris2$Species = factor(iris$Species == "versicolor")
iris.cf = cforest(Species ~ ., data = iris2,control = cforest_unbiased(mtry = 2, ntree = 50))
set.seed(123)
varImpAUC(object = iris.cf)
```

---

varImpRanger   *varImpRanger*

Description

Computes the variable importance for ranger models and for arbitrary measures from the 'measures' package.

Usage

```
varImpRanger(object, data, target, nperm = 1, measure = "multiclass.Brier")
```

Arguments

object	An object as returned by cforest. ranger with option keep.inbag = TRUE.
data	Original data that was used for training the random forest.
target	Target variable as used in the trained model.
nperm	The number of permutations performed.
measure	The name of the measure of the 'measures' package that should be used for the variable importance calculation.

Value

Vector with computed permutation importance for each variable.

Examples

```
## Not run:
library(ranger)
iris.rg = ranger(Species ~ ., data = iris, keep.inbag = TRUE, probability = TRUE)
vimp.ranger = varImpRanger(object = iris.rg, data = iris, target = "Species")
vimp.ranger

## End(Not run)
```

Index