Package ‘IPMRF’

August 9, 2017

Type Package

Title Intervention in Prediction Measure (IPM) for Random Forests

Version 1.2

Date 2017-08-09

Author Irene Epifanio, Stefano Nembrini

Maintainer Irene Epifanio <epifanio@uji.es>

Imports party, randomForest, gbm

Suggests mlbench, randomForestSRC, ranger

Description Computes IPM for assessing variable importance for random forests. See details at I. Epifanio (2017) <DOI:10.1186/s12859-017-1650-8>.

License GPL-3

NeedsCompilation no

Repository CRAN

Date/Publication 2017-08-09 12:14:05 UTC

R topics documented:

`IPMRF-package` *Intervention in Prediction Measure (IPM) for Random Forests*

Description

It computes IPM for assessing variable importance for random forests. See I. Epifanio (2017). Intervention in prediction measure: a new approach to assessing variable importance for random forests. BMC Bioinformatics.

Details

Package:	IPMRF
Type:	Package
Version:	1.2
Date:	2017-08-09

Main Functions:

- ipmparty: IPM casewise with CIT-RF by **party** for OOB samples
- ipmpartynew: IPM casewise with CIT-RF by **party** for new samples
- ipmrf: IPM casewise with CART-RF by **randomForest** for OOB samples
- ipmrfnew: IPM casewise with CART-RF by **randomForest** for new samples
- ipmranger: IPM casewise with RF by **ranger** for OOB samples
- ipmrangernew: IPM casewise with RF by **ranger** for new samples
- ipmgbmnew: IPM casewise with GBM by **gbm** for new samples

Author(s)

Irene Epifanio, Stefano Nembrini

References

Pierola, A. and Epifanio, I. and Alemany, S. (2016) An ensemble of ordered logistic regression and random forest for child garment size matching. *Computers & Industrial Engineering*, **101**, 455–465.

Epifanio, I. (2017) Intervention in prediction measure: a new approach to assessing variable importance for random forests. *BMC Bioinformatics*, **18**, 230.

See Also

https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-017-1650-8

`ipmgbmnew` — *IPM casewise with gbm object by* **gbm** *for new cases, whose responses do not need to be known*

Description

The IPM of a new case, i.e. one not used to grow the forest and whose true response does not need to be known, is computed as follows. The new case is put down each of the *ntree* trees in the forest. For each tree, the case goes from the root node to a leaf through a series of nodes. The variable split in these nodes is recorded. The percentage of times a variable is selected along the case's way from the root to the terminal node is calculated for each tree. Note that we do not count the percentage of times a split occurred on variable k in tree t, but only the variables that intervened in the prediction of the case. The IPM for this new case is obtained by averaging those percentages over the *ntree* trees.

Usage

```
ipmgbmnew(marbolr, da, ntree)
```

Arguments

`marbolr`	Generalized Boosted Regression object obtained with `gbm`.
`da`	Data frame with the predictors only, not responses, for the new cases. Each row corresponds to an observation and each column corresponds to a predictor, which obviously must be the same variables used as predictors in the training set.
`ntree`	Number of trees.

Details

All details are given in Epifanio (2017).

Value

It returns IPM for new cases. It is a matrix with as many rows as cases are in da, and as many columns as predictors are in da.

Note

See Epifanio (2017) about the parameters of RFs to be used, the advantages and limitations of IPM, and in particular when CART is considered with predictors of different types.

Author(s)

Stefano Nembrini

References

Pierola, A. and Epifanio, I. and Alemany, S. (2016) An ensemble of ordered logistic regression and random forest for child garment size matching. *Computers & Industrial Engineering*, **101**, 455–465.

Epifanio, I. (2017) Intervention in prediction measure: a new approach to assessing variable importance for random forests. *BMC Bioinformatics*, **18**, 230.

See Also

ipmparty, ipmrf, ipmranger, ipmpartynew, ipmrfnew

Examples

```
## Not run:
library(party)
library(gbm)
gbm=gbm(score ~ ., data = readingSkills, n.trees=50, shrinkage=0.05, interaction.depth=5,
        bag.fraction = 0.5, train.fraction = 0.5, n.minobsinnode = 1,
        cv.folds = 0, keep.data=F, verbose=F)
apply(ipmgbmnew(gbm,readingSkills[,-4],50),FUN=mean,2)->gbm_ipm
gbm_ipm
## End(Not run)
```

---

ipmparty — *IPM casewise with CIT-RF by* **party** *for OOB samples*

Description

The IPM for a case in the training set is calculated by considering and averaging over only the trees where the case belongs to the OOB set. The case is put down each of the trees where the case belongs to the OOB set. For each tree, the case goes from the root node to a leaf through a series of nodes. The variable split in these nodes is recorded. The percentage of times a variable is selected along the case's way from the root to the terminal node is calculated for each tree. Note that we do not count the percentage of times a split occurred on variable k in tree t, but only the variables that intervened in the prediction of the case. The IPM for this case is obtained by averaging those percentages over only the trees where the case belongs to the OOB set. The random forest is based on CIT (Conditional Inference Trees).

Usage

```
ipmparty(marbol, da, ntree)
```

Arguments

`marbol`	Random forest obtained with `cforest` . Responses can be of the same type supported by `cforest`, not only numerical or nominal, but also ordered responses, censored response variables and multivariate responses.
`da`	Data frame with the predictors only, not responses, of the training set used for computing *marbol*. Each row corresponds to an observation and each column corresponds to a predictor. Predictors can be numeric, nominal or an ordered factor.
`ntree`	Number of trees in the random forest.

Details

All details are given in Epifanio (2017).

Value

It returns IPM for cases in the training set. It is estimated when they are OOB observations. It is a matrix with as many rows as cases are in da, and as many columns as predictors are in da. IPM can be estimated for any kind of RF computed by `cforest`, including multivariate RF.

Note

See Epifanio (2017) about advantages and limitations of IPM, and about the parameters to be used in `cforest`.

Author(s)

Irene Epifanio

References

Pierola, A. and Epifanio, I. and Alemany, S. (2016) An ensemble of ordered logistic regression and random forest for child garment size matching. *Computers & Industrial Engineering*, **101**, 455–465.

Epifanio, I. (2017) Intervention in prediction measure: a new approach to assessing variable importance for random forests. *BMC Bioinformatics*, **18**, 230.

See Also

`ipmpartynew`, `ipmrf`, `ipmranger`, `ipmrfnew`, `ipmrangernew`, `ipmgbmnew`

Examples

```
#Note: more examples can be found at
#https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-017-1650-8

## -------------------------------------------------------
## Example from \code{\link[party]{varimp}} in \pkg{party}
## Classification RF
```

```
## -----------------------------------------------------

## Not run:
library(party)

#from help in varimp by party package
set.seed(290875)
readingSkills.cf <- cforest(score ~ ., data = readingSkills,
control = cforest_unbiased(mtry = 2, ntree = 50))

# standard importance
varimp(readingSkills.cf)

# the same modulo random variation
varimp(readingSkills.cf, pre1.0_0 = TRUE)

# conditional importance, may take a while...
varimp(readingSkills.cf, conditional = TRUE)

## End(Not run)

#IMP based on CIT-RF (party package)
library(party)

ntree<-50
#readingSkills: data from party package
da<-readingSkills[,1:3]
set.seed(290875)
readingSkills.cf3 <- cforest(score ~ ., data = readingSkills,
control = cforest_unbiased(mtry = 3, ntree = 50))

#IPM case-wise computed with OOB with party
pupf<-ipmparty(readingSkills.cf3 ,da,ntree)

#global IPM
pua<-apply(pupf,2,mean)
pua

## -----------------------------------------------------
## Example from \code{\link[randomForestSRC]{var.select}} in \pkg{randomForestSRC}
## Multivariate mixed forests
## -----------------------------------------------------

## Not run:
library(randomForestSRC)

#from help in var.select by randomForestSRC package
mtcars.new <- mtcars
mtcars.new$cyl <- factor(mtcars.new$cyl)
mtcars.new$carb <- factor(mtcars.new$carb, ordered = TRUE)
mv.obj <- rfsrc(cbind(carb, mpg, cyl) ~., data = mtcars.new,
importance = TRUE)
var.select(mv.obj, method = "vh.vimp", nrep = 10)
```

```
#different variables are selected if var.select is repeated


## End(Not run)

#IMP based on CIT-RF (party package)
library(randomForestSRC)
mtcars.new <- mtcars

ntree<-500
da<-mtcars.new[,3:10]
mc.cf <- cforest(carb+ mpg+ cyl ~., data = mtcars.new,
control = cforest_unbiased(mtry = 8, ntree = 500))

#IPM case-wise computing with OOB with party
pupf<-ipmparty(mc.cf ,da,ntree)

#global IPM
pua<-apply(pupf,2,mean)
pua

#disp and hp are consistently selected as more important if repeated
```

`ipmpartynew` *IPM casewise with CIT-RF by* **party** *for new cases, whose responses do not need to be known*

Description

The IPM of a new case, i.e. one not used to grow the forest and whose true response does not need to be known, is computed as follows. The new case is put down each of the *ntree* trees in the forest. For each tree, the case goes from the root node to a leaf through a series of nodes. The variable split in these nodes is recorded. The percentage of times a variable is selected along the case's way from the root to the terminal node is calculated for each tree. Note that we do not count the percentage of times a split occurred on variable k in tree t, but only the variables that intervened in the prediction of the case. The IPM for this new case is obtained by averaging those percentages over the *ntree* trees. The random forest is based on CIT (Conditional Inference Trees).

Usage

```
ipmpartynew(marbol, da, ntree)
```

Arguments

`marbol` Random forest obtained with `cforest` . Responses in the training set can be of the same type supported by `cforest`, not only numerical or nominal, but also ordered responses, censored response variables and multivariate responses.

da	Data frame with the predictors only, not responses, for the new cases. Each row corresponds to an observation and each column corresponds to a predictor, which obviously must be the same variables used as predictors in the training set. Predictors can be numeric, nominal or an ordered factor.
ntree	Number of trees in the random forest.

Details

All details are given in Epifanio (2017).

Value

It returns IPM for new cases. It is a matrix with as many rows as cases are in da, and as many columns as predictors are in da. IPM can be estimated for any kind of RF computed by `cforest`, including multivariate RF.

Note

See Epifanio (2017) about advantages and limitations of IPM, and about the parameters to be used in `cforest`.

Author(s)

Irene Epifanio

References

Pierola, A. and Epifanio, I. and Alemany, S. (2016) An ensemble of ordered logistic regression and random forest for child garment size matching. *Computers & Industrial Engineering*, **101**, 455–465.

Epifanio, I. (2017) Intervention in prediction measure: a new approach to assessing variable importance for random forests. *BMC Bioinformatics*, **18**, 230.

See Also

`ipmparty`, `ipmrf`, `ipmranger`, `ipmrfnew`, `ipmrangernew`, `ipmgbmnew`

Examples

```
#Note: more examples can be found at
#https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-017-1650-8

## -------------------------------------------------------
## Example from \code{\link[party]{varimp}} in \pkg{party}
## Classification RF
## -------------------------------------------------------


library(party)
```

```
#IMP based on CIT-RF (party package)
ntree=50
#readingSkills: data from party package
da=readingSkills[,1:3]
set.seed(290875)
readingSkills.cf3 <- cforest(score ~ ., data = readingSkills,
control = cforest_unbiased(mtry = 3, ntree = 50))

#new case
nativeSpeaker='yes'
age=8
shoeSize=28
da1=data.frame(nativeSpeaker, age, shoeSize)

#IPM case-wise computed for new cases for party package
pupfn=ipmpartynew(readingSkills.cf3,da1,ntree)
pupfn
```

`ipmranger` *IPM casewise with RF by* **ranger** *for OOB samples*

Description

The IPM for a case in the training set is calculated by considering and averaging over only the trees where the case belongs to the OOB set. The case is put down each of the trees where the case belongs to the OOB set. For each tree, the case goes from the root node to a leaf through a series of nodes. The variable split in these nodes is recorded. The percentage of times a variable is selected along the case's way from the root to the terminal node is calculated for each tree. Note that we do not count the percentage of times a split occurred on variable k in tree t, but only the variables that intervened in the prediction of the case. The IPM for this case is obtained by averaging those percentages over only the trees where the case belongs to the OOB set. The random forest is based on a fast implementation of CART-RF.

Usage

```
ipmranger(marbolr, da, ntree)
```

Arguments

`marbolr`	Random forest obtained with `ranger`. Responses can be of the same type supported by `ranger`. Note that not only numerical or nominal, but also ordered responses, censored response variables and multivariate responses can be considered with `ipmparty`.
`da`	Data frame with the predictors only, not responses, of the training set used for computing *marbolr*. Each row corresponds to an observation and each column corresponds to a predictor. Predictors can be numeric, nominal or an ordered factor.
`ntree`	Number of trees in the random forest.

Details

All details are given in Epifanio (2017).

Value

It returns IPM for cases in the training set. It is estimated when they are OOB observations. It is a matrix with as many rows as cases are in da, and as many columns as predictors are in da.

Note

See Epifanio (2017) about the parameters of RFs to be used, the advantages and limitations of IPM, and in particular when CART is considered with predictors of different types.

Author(s)

Stefano Nembrini, Irene Epifanio

References

Pierola, A. and Epifanio, I. and Alemany, S. (2016) An ensemble of ordered logistic regression and random forest for child garment size matching. *Computers & Industrial Engineering*, **101**, 455–465.

Epifanio, I. (2017) Intervention in prediction measure: a new approach to assessing variable importance for random forests. *BMC Bioinformatics*, **18**, 230.

See Also

ipmparty, ipmrf, ipmpartynew, ipmrfnew, ipmrangernew, ipmgbmnew

Examples

```
#Note: more examples can be found at
#https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-017-1650-8

## Not run:
library(ranger)
num.trees=500
rf <- ranger(Species ~ ., data = iris,keep.inbag = TRUE,num.trees=num.trees)

IPM=apply(ipmranger(rf,iris[,-5],num.trees),FUN=mean,2)

## End(Not run)
```

`ipmrangernew` *IPM casewise with RF by* **ranger** *for new cases, whose responses do not need to be known*

Description

The IPM of a new case, i.e. one not used to grow the forest and whose true response does not need to be known, is computed as follows. The new case is put down each of the *ntree* trees in the forest. For each tree, the case goes from the root node to a leaf through a series of nodes. The variable split in these nodes is recorded. The percentage of times a variable is selected along the case's way from the root to the terminal node is calculated for each tree. Note that we do not count the percentage of times a split occurred on variable k in tree t, but only the variables that intervened in the prediction of the case. The IPM for this new case is obtained by averaging those percentages over the *ntree* trees.

The random forest is based on a fast implementation of CART.

Usage

```
ipmrangernew(marbolr, da, ntree)
```

Arguments

`marbolr`	Random forest obtained with `ranger`. Responses can be of the same type supported by `ranger`. Note that not only numerical or nominal, but also ordered responses, censored response variables and multivariate responses can be considered with `ipmparty`.
`da`	Data frame with the predictors only, not responses, for the new cases. Each row corresponds to an observation and each column corresponds to a predictor, which obviously must be the same variables used as predictors in the training set.
`ntree`	Number of trees in the random forest.

Details

All details are given in Epifanio (2017).

Value

It returns IPM for new cases. It is a matrix with as many rows as cases are in da, and as many columns as predictors are in da.

Note

See Epifanio (2017) about the parameters of RFs to be used, the advantages and limitations of IPM, and in particular when CART is considered with predictors of different types.

Author(s)

Stefano Nembrini, Irene Epifanio

References

Pierola, A. and Epifanio, I. and Alemany, S. (2016) An ensemble of ordered logistic regression and random forest for child garment size matching. *Computers & Industrial Engineering*, **101**, 455–465.

Epifanio, I. (2017) Intervention in prediction measure: a new approach to assessing variable importance for random forests. *BMC Bioinformatics*, **18**, 230.

See Also

ipmparty, ipmrf, ipmranger, ipmpartynew, ipmrfnew, ipmgbmnew

Examples

```
## Not run:
library(ranger)
num.trees=500
rf <- ranger(Species ~ ., data = iris,keep.inbag = TRUE,num.trees=num.trees)

IPM_complete=apply(ipmrangernew(rf,iris[,-5],num.trees),FUN=mean,2)

## End(Not run)
```

---

ipmrf — *IPM casewise with CART-RF by* **randomForest** *for OOB samples*

Description

The IPM for a case in the training set is calculated by considering and averaging over only the trees where the case belongs to the OOB set. The case is put down each of the trees where the case belongs to the OOB set. For each tree, the case goes from the root node to a leaf through a series of nodes. The variable split in these nodes is recorded. The percentage of times a variable is selected along the case's way from the root to the terminal node is calculated for each tree. Note that we do not count the percentage of times a split occurred on variable k in tree t, but only the variables that intervened in the prediction of the case. The IPM for this case is obtained by averaging those percentages over only the trees where the case belongs to the OOB set. The random forest is based on CART.

Usage

```
ipmrf(marbolr, da, ntree)
```

Arguments

`marbolr` Random forest obtained with `randomForest`. Responses can be of the same type supported by `randomForest`. Note that not only numerical or nominal, but also ordered responses, censored response variables and multivariate responses can be considered with `ipmparty`.

`da` Data frame with the predictors only, not responses, of the training set used for computing *marbolr*. Each row corresponds to an observation and each column corresponds to a predictor. Predictors can be numeric, nominal or an ordered factor.

`ntree` Number of trees in the random forest.

Details

All details are given in Epifanio (2017).

Value

It returns IPM for cases in the training set. It is estimated when they are OOB observations. It is a matrix with as many rows as cases are in da, and as many columns as predictors are in da.

Note

See Epifanio (2017) about the parameters of RFs to be used, the advantages and limitations of IPM, and in particular when CART is considered with predictors of different types.

Author(s)

Irene Epifanio

References

Pierola, A. and Epifanio, I. and Alemany, S. (2016) An ensemble of ordered logistic regression and random forest for child garment size matching. *Computers & Industrial Engineering*, **101**, 455–465.

Epifanio, I. (2017) Intervention in prediction measure: a new approach to assessing variable importance for random forests. *BMC Bioinformatics*, **18**, 230.

See Also

`ipmparty`, `ipmranger`, `ipmpartynew`, `ipmrfnew`, `ipmrangernew`, `ipmgbmnew`

Examples

```
#Note: more examples can be found at
#https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-017-1650-8

## Not run:

library(mlbench)
```

```
#data used by Breiman, L.: Random forests. Machine Learning 45(1), 5--32 (2001)
data(PimaIndiansDiabetes2)
Diabetes <- na.omit(PimaIndiansDiabetes2)

set.seed(2016)
require(randomForest)
ri<- randomForest(diabetes  ~ ., data=Diabetes, ntree=500, importance=TRUE,
keep.inbag=TRUE,replace = FALSE)

#GVIM and PVIM (CART-RF)
im=importance(ri)
im
#rank
ii=apply(im,2,rank)
ii

#IPM based on CART-RF (randomForest package)
da=Diabetes[,1:8]
ntree=500
#IPM case-wise computed with OOB
pupf=ipmrf(ri,da,ntree)

#global IPM
pua=apply(pupf,2,mean)
pua

#IPM by classes
attach(Diabetes)
puac=matrix(0,nrow=2,ncol=dim(da)[2])
puac[1,]=apply(pupf[diabetes=='neg',],2,mean)
puac[2,]=apply(pupf[diabetes=='pos',],2,mean)
colnames(puac)=colnames(da)
rownames(puac)=c( 'neg', 'pos')
puac

#rank IPM
#global rank
rank(pua)
#rank by class
apply(puac,1,rank)

## End(Not run)
```

---

ipmrfnew *IPM casewise with CART-RF by* **randomForest** *for new cases, whose responses do not need to be known*

Description

The IPM of a new case, i.e. one not used to grow the forest and whose true response does not need to be known, is computed as follows. The new case is put down each of the *ntree* trees in the forest.

For each tree, the case goes from the root node to a leaf through a series of nodes. The variable split in these nodes is recorded. The percentage of times a variable is selected along the case's way from the root to the terminal node is calculated for each tree. Note that we do not count the percentage of times a split occurred on variable k in tree t, but only the variables that intervened in the prediction of the case. The IPM for this new case is obtained by averaging those percentages over the *ntree* trees.

The random forest is based on CART

Usage

```
ipmrfnew(marbolr, da, ntree)
```

Arguments

`marbolr`	Random forest obtained with `randomForest`. Responses can be of the same type supported by `randomForest`. Note that not only numerical or nominal, but also ordered responses, censored response variables and multivariate responses can be considered with `ipmparty`.
`da`	Data frame with the predictors only, not responses, for the new cases. Each row corresponds to an observation and each column corresponds to a predictor, which obviously must be the same variables used as predictors in the training set.
`ntree`	Number of trees in the random forest.

Details

All details are given in Epifanio (2017).

Value

It returns IPM for new cases. It is a matrix with as many rows as cases are in da, and as many columns as predictors are in da.

Note

See Epifanio (2017) about the parameters of RFs to be used, the advantages and limitations of IPM, and in particular when CART is considered with predictors of different types.

Author(s)

Irene Epifanio

References

Pierola, A. and Epifanio, I. and Alemany, S. (2016) An ensemble of ordered logistic regression and random forest for child garment size matching. *Computers & Industrial Engineering*, **101**, 455–465.

Epifanio, I. (2017) Intervention in prediction measure: a new approach to assessing variable importance for random forests. *BMC Bioinformatics*, **18**, 230.

See Also

ipmparty, ipmrf, ipmranger, ipmpartynew, ipmrangernew,ipmgbmnew

Examples

```
#Note: more examples can be found at
#https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-017-1650-8


library(mlbench)
#data used by Breiman, L.: Random forests. Machine Learning 45(1), 5--32 (2001)
data(PimaIndiansDiabetes2)
Diabetes <- na.omit(PimaIndiansDiabetes2)

set.seed(2016)
require(randomForest)
ri<- randomForest(diabetes  ~ ., data=Diabetes, ntree=500, importance=TRUE,
keep.inbag=TRUE,replace = FALSE)

#new cases
da1=rbind(apply(Diabetes[Diabetes[,9]=='pos',1:8],2,mean),
apply(Diabetes[Diabetes[,9]=='neg',1:8],2,mean))


#IPM case-wise computed for new cases for randomForest package
ntree=500
pupfn=ipmrfnew(ri, as.data.frame(da1),ntree)
pupfn
```

Index