Package 'CSMES'

February 3, 2023

Type Package

Title Cost-Sensitive Multi-Criteria Ensemble Selection for Uncertain Cost Conditions

Version 1.0.1

Author Koen W. De Bock, Kristof Coussement and Stefan Lessmann

Maintainer Koen W. De Bock <kdebock@audencia.com>

Description Functions for cost-sensitive multi-criteria ensemble selection (CSMES) (as described in De bock et al. (2020) <doi:10.1016/j.ejor.2020.01.052>) for cost-sensitive learning under unknown cost conditions.

License GPL (>= 2)

Imports mco (>= 1.0-15.1), ROCR (>= 1.0-7), rpart (>= 4.1-15), zoo (>= 1.8-6), graphics (>= 3.5.1), stats (>= 3.5.1), caTools (>= 1.18.0), data.table (>= 1.12.2)

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

NeedsCompilation no

Repository CRAN

Date/Publication 2023-02-03 14:02:31 UTC

R topics documented:

BFP *Business failure prediction demonstration data set*

Description

Business failure prediction demonstration data set. Contains financial ratios and firmographics as independent variables for 522 anonymized European companies. The Class column indicates failure (class 1) or survival (class 0) over a 1-year period.

Author(s)

Koen W. De Bock, `<kdebock@audencia.com>`

References

De Bock, K.W., Lessmann, S. And Coussement, K., Cost-sensitive business failure prediction when misclassification costs are uncertain: A heterogeneous ensemble selection approach, European Journal of Operational Research (2020), doi: 10.1016/j.ejor.2020.01.052.

brierCurve *Calculates Brier Curve*

Description

This function calculates the Brier curve (both in terms of cost and skew) based on a set of predictions generated by a binary classifier. Brier curves allow an evaluation of classifier performance in cost space. This code is an adapted version from the authors' original implementation, available through http://dmip.webs.upv.es/BrierCurves/BrierCurves.R.

Usage

```
brierCurve(labels, preds, resolution = 0.001)
```

Arguments

`labels`	Vector with true class labels
`preds`	Vector with predictions (real-valued or discrete)
`resolution`	Value for the determination of percentile intervals. Defaults to 1/1000.

Value

object of the class `brierCurve` which is a list with the following components:

`brierCurveCost` Cost-based Brier curve, represented as (cost,loss) coordinates

`brierCurveSkew` Skew-based Brier curve, represented as (skew,loss) coordinates

`auc_brierCurveCost`
Area under the cost-based Brier curve.

`auc_brierCurveSkew`
Area under the skew-based Brier curve.

Author(s)

Koen W. De Bock, <kdebock@audencia.com>

References

Hernandez-Orallo, J., Flach, P., & Ferri, C. (2011). Brier Curves: a New Cost-Based Visualisation of Classifier Performance. Proceedings of the 28th International Conference on Machine Learning (ICML-11), 585–592.

See Also

`plotBrierCurve`, `CSMES.ensNomCurve`

Examples

```
##load data
library(rpart)
data(BFP)
##generate random order vector
BFP_r<-BFP[sample(nrow(BFP),nrow(BFP)),]
size<-nrow(BFP_r)
##size<-300
train<-BFP_r[1:floor(size/3),]
val<-BFP_r[ceiling(size/3):floor(2*size/3),]
test<-BFP_r[ceiling(2*size/3):size,]
##train CART decision tree model
model=rpart(as.formula(Class~.),train,method="class")
##generate predictions for the tes set
preds<-predict(model,newdata=test)[,2]
##calculate brier curve
bc<-brierCurve(test[,"Class"],preds)
```

`CSMES.ensNomCurve` *CSMES Training Stage 2: Extract an ensemble nomination curve (cost curve- or Brier curve-based) from a set of Pareto-optimal ensemble classifiers*

Description

Generates an ensemble nomination curve from a set of Pareto-optimal ensemble definitions as identified through `CSMES.ensSel`).

Usage

```
CSMES.ensNomCurve(
  ensSelModel,
  memberPreds,
  y,
  curveType = c("costCurve", "brierSkew", "brierCost"),
  method = c("classPreds", "probPreds"),
  plotting = FALSE,
  nrBootstraps = 1
)
```

Arguments

`ensSelModel`	ensemble selection model (output of `CSMES.ensSel`)
`memberPreds`	matrix containing ensemble member library predictions
`y`	Vector with true class labels. Currently, a dichotomous outcome variable is supported
`curveType`	the type of cost curve used to construct the ensemble nomination curve. Shoul be "brierCost","brierSkew" or "costCurve" (default).
`method`	how are candidate ensemble learner predictions used to generate the ensemble nomination front? "classPreds" for class predictions (default), "probPreds" for probability predictions.
`plotting`	`TRUE` or `FALSE`: Should a plot be generated showing the Brier curve? Defaults to `FALSE`.
`nrBootstraps`	optionally, the ensemble nomination curve can be generated through bootstrapping. This argument specifies the number of iterations/bootstrap samples. Default is 1.

Value

An object of the class `CSMES.ensNomCurve` which is a list with the following components:

`nomcurve`	the ensemble nomination curve
`curves`	individual cost curves or brier curves of ensemble members

`intervals` resolution of the ensemble nomination curve

`incidence` incidence (positive rate) of the outcome variable

`area_under_curve` area under the ensemble nomination curve

`method` method used to generate the ensemble nomination front:"classPreds" for class predictions (default), "probPreds" for probability predictions

`curveType` the type of cost curve used to construct the ensemble nomination curve

`nrBootstraps` number of boostrap samples over which the ensemble nomination curve was estimated

Author(s)

Koen W. De Bock, <kdebock@audencia.com>

References

De Bock, K.W., Lessmann, S. And Coussement, K., Cost-sensitive business failure prediction when misclassification costs are uncertain: A heterogeneous ensemble selection approach, European Journal of Operational Research (2020), doi: 10.1016/j.ejor.2020.01.052.

See Also

CSMES.ensSel, CSMES.predictPareto, CSMES.predict

Examples

```
##load data
library(rpart)
library(zoo)
library(ROCR)
library(mco)
data(BFP)
##generate random order vector
BFP_r<-BFP[sample(nrow(BFP),nrow(BFP)),]
size<-nrow(BFP_r)
##size<-300
train<-BFP_r[1:floor(size/3),]
val<-BFP_r[ceiling(size/3):floor(2*size/3),]
test<-BFP_r[ceiling(2*size/3):size,]
##generate a list containing model specifications for 100 CART decisions trees varying in the cp
##and minsplit parameters, and trained on bootstrap samples (bagging)
rpartSpecs<-list()
for (i in 1:100){
  data<-train[sample(1:ncol(train),size=ncol(train),replace=TRUE),]
  str<-paste("rpartSpecs$rpart",i,"=rpart(as.formula(Class~.),data,method=\"class\",
  control=rpart.control(minsplit=",round(runif(1, min = 1, max = 20)),",cp=",runif(1,
  min = 0.05, max = 0.4),"))",sep="")
  eval(parse(text=str))
}
##generate predictions for these models
```

```
hillclimb<-mat.or.vec(nrow(val),100)
for (i in 1:100){
  str<-paste("hillclimb[,",i,"]=predict(rpartSpecs[[i]],newdata=val)[,2]",sep="")
  eval(parse(text=str))
}
##score the validation set used for ensemble selection, to be used for ensemble selection
ESmodel<-CSMES.ensSel(hillclimb,val$Class,obj1="FNR",obj2="FPR",selType="selection",
generations=10,popsize=12,plot=TRUE)
## Create Ensemble nomination curve
enc<-CSMES.ensNomCurve(ESmodel,hillclimb,val$Class,curveType="costCurve",method="classPreds",
plot=FALSE)
```

`CSMES.ensSel` *CSMES Training Stage 1: Cost-Sensitive Multicriteria Ensemble Selection resulting in a Pareto frontier of candidate ensemble classifiers*

Description

This function applies the first stage in the learning process of CSMES: optimizing Cost-Sensitive Multicriteria Ensemble Selection, resulting in a Pareto frontier of equivalent candidate ensemble classifiers along two objective functions. By default, cost space is optimized by optimizing false positive and false negative rates simultaneously. This results in a set of optimal ensemble classifiers, varying in the tradeoff between FNR and FPR. Optionally, other objective metrics can be specified. Currently, only binary classification is supported.

Usage

```
CSMES.ensSel(
  memberPreds,
  y,
  obj1 = c("FNR", "AUCC", "MSE", "AUC"),
  obj2 = c("FPR", "ensSize", "ensSizeSq", "clAmb"),
  selType = c("selection", "selectionWeighted", "weighted"),
  plotting = TRUE,
  generations = 30,
  popsize = 100
)
```

Arguments

`memberPreds`	matrix containing ensemble member library predictions
`y`	Vector with true class labels. Currently, a dichotomous outcome variable is supported
`obj1`	Specifies the first objective metric to be minimized
`obj2`	Specifies the second objective metric to be minimized
`selType`	Specifies the type of ensemble selection to be applied: `"selection"` for basic selection, `"selectionWeighted"` for weighted selection, `"weighted"` for weighted sum

`plotting`	`TRUE` or `FALSE`: Should a plot be generated showing objective function values throughout the optimization process?
`generations`	the number of population generations for nsga-II. Default is 30.
`popsize`	the population size for nsga-II. Default is 100.

Value

An object of the class `CSMES.ensSel` which is a list with the following components:

`weights`	ensemble member weights for all pareto-optimal ensemble classifiers after multicriteria ensemble selection
`obj_values`	optimization objective values
`pareto`	overview of pareto-optimal ensemble classifiers
`popsize`	the population size for nsga-II
`generarations`	the number of population generations for nsga-II
`obj1`	Specifies the first objective metric that was minimized
`obj2`	Specifies the second objective metric that was minimized
`selType`	the type of ensemble selection that was applied: `"selection"`, `"selectionWeighted"` or `"weighted"`
`ParetoPredictions_p`	probability predictions for pareto-optimal ensemble classifiers
`ParetoPredictions_c`	class predictions for pareto-optimal ensebmle classifiers

Author(s)

Koen W. De Bock, `<kdebock@audencia.com>`

References

De Bock, K.W., Lessmann, S. And Coussement, K., Cost-sensitive business failure prediction when misclassification costs are uncertain: A heterogeneous ensemble selection approach, European Journal of Operational Research (2020), doi: 10.1016/j.ejor.2020.01.052.

Examples

```
##load data
library(rpart)
library(zoo)
library(ROCR)
library(mco)
data(BFP)
##generate random order vector
BFP_r<-BFP[sample(nrow(BFP),nrow(BFP)),]
size<-nrow(BFP_r)
##size<-300
train<-BFP_r[1:floor(size/3),]
val<-BFP_r[ceiling(size/3):floor(2*size/3),]
```

```
test<-BFP_r[ceiling(2*size/3):size,]
##generate a list containing model specifications for 100 CART decisions trees varying in the cp
##and minsplit parameters, and trained on bootstrap samples (bagging)
rpartSpecs<-list()
for (i in 1:100){
  data<-train[sample(1:ncol(train),size=ncol(train),replace=TRUE),]
  str<-paste("rpartSpecs$rpart",i,"=rpart(as.formula(Class~.),data,method=\"class\",
  control=rpart.control(minsplit=",round(runif(1, min = 1, max = 20)),",cp=",runif(1,
  min = 0.05, max = 0.4),"))",sep="")
  eval(parse(text=str))
}
##generate predictions for these models
hillclimb<-mat.or.vec(nrow(val),100)
for (i in 1:100){
  str<-paste("hillclimb[,",i,"]=predict(rpartSpecs[[i]],newdata=val)[,2]",sep="")
  eval(parse(text=str))
}
##score the validation set used for ensemble selection, to be used for ensemble selection
ESmodel<-CSMES.ensSel(hillclimb,val$Class,obj1="FNR",obj2="FPR",selType="selection",
generations=10,popsize=12,plot=TRUE)
## Create Ensemble nomination curve
enc<-CSMES.ensNomCurve(ESmodel,hillclimb,val$Class,curveType="costCurve",method="classPreds",
plot=FALSE)
```

`CSMES.predict` *CSMES scoring: generate predictions for the optimal ensemble classifier according to CSMES in function of cost information.*

Description

This function generates predictions for a new data set (containing candidate member library predictions) using a CSMES model. Using Pareto-optimal ensemble definitions generated through `CSMES.ensSel` and the ensemble nomination front generated using `CSMES.EnsNomCurve`, final ensemble predictions are generated in function of cost information known to the user at the time of model scoring. The model allows for three scenarios: (1) the candidate ensemble is nominated in function of a specific cost ratio, (2) the ensemble is nominated in function of partial AUCC (or a distribution over operating points) and (3) the candidate ensemble that is optimal over the entire cost space in function of area under the cost or brier curve is chosen.

Usage

```
CSMES.predict(
  ensSelModel,
  ensNomCurve,
  newdata,
  criterion = c("minEMC", "minAUCC", "minPartAUCC"),
  costRatio = 5,
  partAUCC_mu = 0.5,
  partAUCC_sd = 0.1
)
```

Arguments

`ensSelModel`	ensemble selection model (output of `CSMES.ensSel`)
`ensNomCurve`	ensemble nomination curve object (output of `CSMES.ensNomCurve`)
`newdata`	matrix containing ensemble library member model predictions for new data set
`criterion`	This argument specifies which criterion determines the selection of the ensemble candidate that delivers predictions. Can be one of three options: "minEMC", "minAUCC" or "minPartAUCC".
`costRatio`	Specifies the cost ratio used to determine expected misclassification cost. Only relvant when `criterion` is "minEMC".
`partAUCC_mu`	Desired mean operating condition when `criterion` is "minPartAUCC" (partial area under the cost/brier curve).
`partAUCC_sd`	Desired standard deviation when `criterion` is "minPartAUCC" (partial area under the cost/brier curve).

Value

An list with the following components:

`pred`	A matrix with model predictions. Both class and probability predictions are delivered.
`criterion`	The criterion specified to determine the selection of the ensemble candidate.
`costRatio`	The cost ratio in function of which the `criterion` "minEMC" has selected the optimal candidate ensemble that delivered predictions

Author(s)

Koen W. De Bock, `<kdebock@audencia.com>`

References

De Bock, K.W., Lessmann, S. And Coussement, K., Cost-sensitive business failure prediction when misclassification costs are uncertain: A heterogeneous ensemble selection approach, European Journal of Operational Research (2020), doi: 10.1016/j.ejor.2020.01.052.

See Also

`CSMES.ensSel`, `CSMES.predictPareto`, `CSMES.ensNomCurve`

Examples

```
##load data
library(rpart)
library(zoo)
library(ROCR)
library(mco)
data(BFP)
##generate random order vector
BFP_r<-BFP[sample(nrow(BFP),nrow(BFP)),]
```

```
size<-nrow(BFP_r)
##size<-300
train<-BFP_r[1:floor(size/3),]
val<-BFP_r[ceiling(size/3):floor(2*size/3),]
test<-BFP_r[ceiling(2*size/3):size,]
##generate a list containing model specifications for 100 CART decisions trees varying in the cp
##and minsplit parameters, and trained on bootstrap samples (bagging)
rpartSpecs<-list()
for (i in 1:100){
  data<-train[sample(1:ncol(train),size=ncol(train),replace=TRUE),]
  str<-paste("rpartSpecs$rpart",i,"=rpart(as.formula(Class~.),data,method=\"class\",
  control=rpart.control(minsplit=",round(runif(1, min = 1, max = 20)),",cp=",runif(1,
  min = 0.05, max = 0.4),"))",sep="")
  eval(parse(text=str))
}
##generate predictions for these models
hillclimb<-mat.or.vec(nrow(val),100)
for (i in 1:100){
  str<-paste("hillclimb[,",i,"]=predict(rpartSpecs[[i]],newdata=val)[,2]",sep="")
  eval(parse(text=str))
}
##score the validation set used for ensemble selection, to be used for ensemble selection
ESmodel<-CSMES.ensSel(hillclimb,val$Class,obj1="FNR",obj2="FPR",selType="selection",
generations=10,popsize=12,plot=TRUE)
## Create Ensemble nomination curve
enc<-CSMES.ensNomCurve(ESmodel,hillclimb,val$Class,curveType="costCurve",method="classPreds",
plot=FALSE)
```

`CSMES.predictPareto` *Generate predictions for all Pareto-optimal ensemble classifier candidates selected through CSMES*

Description

This function generates predictions for all pareto-optimal ensemble classifier candidates as identified through the first training stage of CSMES (`CSMES.ensSel`).

Usage

```
CSMES.predictPareto(ensSelModel, newdata)
```

Arguments

`ensSelModel`	ensemble selection model (output of `CSMES.ensSel`)
`newdata`	data.frame or matrix containing data to be scored

Value

An object of the class CSMES.predictPareto which is a list with the following two components:

Pareto_predictions_c
: A vector with class predictions.

Paret_predictions_p
: A vector with probability predictions.

Author(s)

Koen W. De Bock, <kdebock@audencia.com>

References

De Bock, K.W., Lessmann, S. And Coussement, K., Cost-sensitive business failure prediction when misclassification costs are uncertain: A heterogeneous ensemble selection approach, European Journal of Operational Research (2020), doi: 10.1016/j.ejor.2020.01.052.

See Also

CSMES.ensSel, CSMES.predict, CSMES.ensNomCurve

Examples

```
##load data
library(rpart)
library(zoo)
library(ROCR)
library(mco)
data(BFP)
##generate random order vector
BFP_r<-BFP[sample(nrow(BFP),nrow(BFP)),]
size<-nrow(BFP_r)
##size<-300
train<-BFP_r[1:floor(size/3),]
val<-BFP_r[ceiling(size/3):floor(2*size/3),]
test<-BFP_r[ceiling(2*size/3):size,]
##generate a list containing model specifications for 100 CART decisions trees varying in the cp
##and minsplit parameters, and trained on bootstrap samples (bagging)
rpartSpecs<-list()
for (i in 1:100){
  data<-train[sample(1:ncol(train),size=ncol(train),replace=TRUE),]
  str<-paste("rpartSpecs$rpart",i,"=rpart(as.formula(Class~.),data,method=\"class\",
  control=rpart.control(minsplit=",round(runif(1, min = 1, max = 20)),",cp=",runif(1,
  min = 0.05, max = 0.4),"))",sep="")
  eval(parse(text=str))
}
##generate predictions for these models
hillclimb<-mat.or.vec(nrow(val),100)
for (i in 1:100){
  str<-paste("hillclimb[,",i,"]=predict(rpartSpecs[[i]],newdata=val)[,2]",sep="")
```

```
  eval(parse(text=str))
}
##score the validation set used for ensemble selection, to be used for ensemble selection
ESmodel<-CSMES.ensSel(hillclimb,val$Class,obj1="FNR",obj2="FPR",selType="selection",
generations=10,popsize=12,plot=TRUE)
## Create Ensemble nomination curve
enc<-CSMES.ensNomCurve(ESmodel,hillclimb,val$Class,curveType="costCurve",method="classPreds",
plot=FALSE)
```

`plotBrierCurve` *Plots Brier Curve*

Description

This function plots the brier curve based on a set of predictions generated by a binary classifier. Brier curves allow an evaluation of classifier performance in cost space.

Usage

```
plotBrierCurve(bc, curveType = c("brierCost", "brierSkew"))
```

Arguments

`bc`	A `brierCurve` object created by the `brierCurve` function
`curveType`	the type of Brier curve to be plotted. Shoul be "brierCost" or"brierSkew".

Value

None

Author(s)

Koen W. De Bock, <kdebock@audencia.com>

References

Hernandez-Orallo, J., Flach, P., & Ferri, C. (2011). Brier Curves: a New Cost-Based Visualisation of Classifier Performance. Proceedings of the 28th International Conference on Machine Learning (ICML-11), 585–592.

See Also

`brierCurve`, `CSMES.ensNomCurve`

Examples

```
##load data
library(rpart)
data(BFP)
##generate random order vector
BFP_r<-BFP[sample(nrow(BFP),nrow(BFP)),]
size<-nrow(BFP_r)
##size<-300
train<-BFP_r[1:floor(size/3),]
val<-BFP_r[ceiling(size/3):floor(2*size/3),]
test<-BFP_r[ceiling(2*size/3):size,]
##train CART decision tree model
model=rpart(as.formula(Class~.),train,method="class")
##generate predictions for the tes set
preds<-predict(model,newdata=test)[,2]
##calculate brier curve
bc<-brierCurve(test[,"Class"],preds)
##plot briercurve
plotBrierCurve(bc,curveType="cost")
```

Index