Package 'EPX'

October 12, 2022

Type Package

Title Ensemble of Phalanxes

Version 1.0.4

Maintainer Jabed Tomal <jtomal@tru.ca>

Description An ensemble method for the statistical detection of
a rare class in two-class classification problems. The method uses an
ensemble of classifiers where the constituent
models of the ensemble use disjoint subsets (phalanxes) of explanatory
variables. We provide an implementation of the phalanx-formation algorithm.
Please see Tomal et al. (2015) <doi:10.1214/14-AOAS778>,
Tomal et al. (2016) <doi:10.1021/acs.jcim.5b00663>, and
Tomal et al. (2019) <arXiv:1706.06971> for more details.

License GPL-3

Encoding UTF-8

LazyData TRUE

Depends R (>= 3.6.0), foreach, randomForest, doRNG

Imports nnet, doParallel, rngtools

RoxygenNote 7.1.1

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation yes

Author Jabed Tomal [aut, cre],
Grace Hsu [aut],
William Welch [aut],
Marcia Wang [ctb]

Repository CRAN

Date/Publication 2021-07-06 21:50:07 UTC

R topics documented:

AHR *Calculate AHR (or expected AHR)*

Description

Calculates average hitrate (AHR), which is equivalent to average precision. When there are ties in ranking (not all values in phat are unique), the result is the expection of AHR. The algorithm that produces this analytic result assumes that the items in any tied group are in an arbitrary order within the group.

Usage

```
AHR(y, phat, ...)
```

Arguments

y	True (binary) response vector where 1 is the rare/relevant class.
phat	Numeric vector of estimated probabilities of relevance.
...	Further arguments passed to or from other methods.

Details

Implementation adapted from Wang (2005, Chapter 3). Please also see Chapter 3 of Wang (2005) for AHR and expected AHR formulas.

Value

Numeric value of average hitrate; expected average hitrate when there are ties.

References

Wang, M. (2005). *Statistical Methods for High Throughput Screening Drug Discovery Data* (Doctoral thesis). University of Waterloo, Waterloo, Ontario, Canada.

Examples

```
## AHR when there are no ties in phat:
resp <- c(1, 0, 0, 0, 1)
prob <- (1:5)*0.1
AHR(y = resp, phat = prob)
# expect answer: 1/2 * (1 + 0 + 0 + 0 + 2/5)

## (Expected) AHR when there are ties in phat:
resp <- c(1, 1, 0,   0,   0,   0,   0,    1,   0, 0)
prob <- c(1, 1, 1, 0.4, 0.4, 0.3, 0.2, 0.15, 0.1, 0)
AHR(y = resp, phat = prob)
# expect answer: 1/3 * (2/3 + 1/2 * (1/3 + 2/3) + 1/3 * 4/3 +
#                       1/8 * (2/3 + 2/3 + 2/3 + 1))
```

`BNhold` *AID348 hold-out data using Burden Numbers for testing the EPX package*

Description

AID348 hold-out data with 24 burden numbers as explanatory variables. Demonstrates in a timely manner `epx`, the phalanx-formation algorithm in **EPX** and associated functions `summary.epx`, `predict.epx` , `plot.epx`, `hit.curve`.

Usage

```
BNhold
```

Format

A dataframe with 3946 rows and 25 variables:

WBN Burden numbers descriptor set with 24 variables.

y The response variable where 1 denotes active and 0 inactive.

References

Tomal, J. H., Welch, W. J., & Zamar, R. H. (2015). Ensembling classification models based on phalanxes of variables with applications in drug discovery. *The Annals of Applied Statistics*, *9*(1), 69-93. doi: 10.1214/14AOAS778

BNsample — *AID348 sample (training) data with Burden Numbers for testing the EPX package*

Description

AID348 sample (training) dataset with 24 burden numbers as explanatory variables. Demonstrates in a timely manner epx, the phalanx-formation algorithm in **EPX** and associated functions summary.epx, predict.epx , plot.epx, cv.epx, hit.curve.

Usage

```
BNsample
```

Format

A dataframe with 1000 rows and 25 variables:

WBN Burden numbers descriptor set with 24 variables.

y The response variable where 1 denotes active and 0 inactive.

References

Tomal, J. H., Welch, W. J., & Zamar, R. H. (2015). Ensembling classification models based on phalanxes of variables with applications in drug discovery. *The Annals of Applied Statistics*, *9*(1), 69-93. doi: 10.1214/14AOAS778

cv.epx — *Balanced K-fold cross-validation for an "epx" object*

Description

Balanced K-fold cross-validation based on an "epx" object. Hence, we have biased cross-validation as we do not re-run the phalanx-formation algorithm for each fold.

Usage

```
cv.epx(
  epx,
  folds = NULL,
  K = 10,
  folds.out = FALSE,
  classifier.args = list(),
  performance.args = list(),
  ...
)
```

Arguments

epx	Object of class "epx".
folds	Optional vector specifying to which fold each observation belongs. Must be an n-length vector (n being the number of observations) with integer values only in the range from 1 to K.
K	Number of folds; default is 10.
folds.out	Indicates whether a vector indicating fold membership for each of the observations will be output; default is FALSE.
classifier.args	Arguments for the base classifier specified by epx; default is that used in epx formation.
performance.args	Arguments for the performance measure specified by epx; default is that used in epx formation.
...	Further arguments passed to or from other methods.

Value

An $(n + 1)$ by $(p + 1)$ matrix, where n is the number of observations used to train epx and p is the number of (final) phalanxes. Column $p + 1$ of the matrix contains the predicted probabilities of relevance from the ensemble of phalanxes, and row $n+1$ is the performance (choice of performance measure determined by the "epx" object) of the corresponding column.

Setting folds.out as TRUE changes the output of cv.epx into a list of two elements:

EPX.CV	The $(n + 1)$ by $(p + 1)$ matrix returned by default when folds.out = FALSE.
FOLDS.USED	A vector of length n with integer values only in the range from 1 to K indicating to which fold each observation was randomly assigned for cross-validation.

Examples

```
# Example with data(harvest)

## Phalanx-formation using a base classifier with 50 trees (default = 500)

set.seed(761)
model <- epx(x = harvest[, -4], y = harvest[, 4],
            classifier.args = list(ntree = 50))

## 10-fold balanced cross-validation (different base classifier settings)
## Not run:
set.seed(761)
cv.100 <- cv.epx(model, classifier.args = list(ntree = 100))
tail(cv.100) # see performance (here, AHR) for all phalanxes and the ensemble


## Option to output the vector assigning observations to the K folds
## (Commented out for speed.)
set.seed(761)
```

```
cv.folds <- cv.epx(model, folds.out = TRUE)
tail(cv.folds[[1]])  # same as first example
table(cv.folds[[2]])  # number of observations in each of the 10 folds

## 10 runs of 10-fold balanced cross-validation (using default settings)
set.seed(761)
cv.ahr <- NULL  # store AHR of each ensemble
for (i in 1:10) {
  cv.i <- cv.epx(model)
  cv.ahr <- c(cv.ahr, cv.i[nrow(cv.i), ncol(cv.i)])
}
boxplot(cv.ahr)  # to see variation in AHR

## End(Not run)
```

---

epx *Fitting an Ensemble of Phalanxes*

Description

epx forms phalanxes of variables from training data for binary classification with a rare class. The phalanxes are disjoint subsets of variables, each of which is fit with a base classifier. Together they form an ensemble.

Usage

```
epx(
  x,
  y,
  phalanxes.initial = c(1:ncol(x)),
  alpha = 0.95,
  nsim = 1000,
  rmin.target = 1,
  classifier = "random forest",
  classifier.args = list(),
  performance = "AHR",
  performance.args = list(),
  computing = "sequential",
  ...
)
```

Arguments

x	Explanatory variables (predictors, features) contained in a data frame.
y	Binary response variable vector (numeric or integer): 1 for the rare class, 0 for the majority class.

`phalanxes.initial`
: Initial variable group indices; default one group per variable. Example: vector c(1, 1, 2, 2, 3, ...) puts variables 1 and 2 in group 1, variables 3 and 4 in group, 2, etc. Indices cannot be skipped, e.g., c(1, 3, 3, 4, 4, 3, 1) skips group 2 and is invalid.

`alpha`
: Lower-tail probability for the critical quantile of the reference distribution of the `performance` measure for a classifier that ranks at random (i.e., the predictors have no explanatory power); default is 0.95.

`nsim`
: Number of simulations for the reference empirical distribution of the performance measure; default is 1000.

`rmin.target`
: To merge the pair of groups with the minimum ratio of performance measures (ensemble of models to single model) into a single group their ratio must be less than `rmin.target`, otherwise merging stops; default is 1.

`classifier`
: Base classifier, one of `c("random forest", "logistic regression", "neural network")`; default is "random forest", which uses `randomForest`.

`classifier.args`
: Arguments for the base `classifier` specified in a list as follows: `list(argName1 = value1, argName2 = value2, ...)`. If the list is empty, the classifier will use its defaults. For "random forest", user may specify `replace, cutoff, nodesize, maxnodes`. For "logistic regression" there are no options. For "neural network", user may specify `size, trace`.

`performance`
: Performance assessment metric, one of `c("AHR", "IE", "TOP1", "RKL")`; default is `AHR`.

`performance.args`
: Arguments for the `performance` measure specified in a list as follows: `list(argName1 = value1, argName2 = value2, ...)`. If the list is empty, the performance measure will use its defaults. Currently, only `IE` takes an argument list, and its only argument is `cutoff`.

`computing`
: Whether to compute sequentially or in parallel. Input is one of `c("sequential", "parallel")`; default is "sequential".

`...`
: Further arguments passed to or from other methods.

Details

Please see Tomal et al. (2015) for more description of phalanx formation.

Value

Returns an object of class `epx`, which is a list containing the following components:

`PHALANXES`
: List of four vectors, each the same length as the number of explanatory variables (columns in `x`): `phalanxes.initial`, `phalanxes.filtered`, `phalanxes.merged`, `phalanxes.final`. Each vector contains the phalanx membership indices of all explanatory variables at one of the four stages of phalanx-formation. Element i of a vector is the index of the phalanx to which variable i belongs. Phalanx 0 does not exist and so membership in phalanx 0 indicates that the variable does not belong to any phalanx; it has been screened out.

PHALANXES.FINAL.PERFORMANCE
: Vector of performance measures of the final phalanxes: the first element is for phalanx 1, etc.

PHALANXES.FINAL.FITS
: A matrix with number of rows equal to the number of observations in the training data and number of columns equal to the number of final phalanxes. Column i contains the predicted probabilities of class 1 from fitting the base classifier to the variables in phalanx i.

ENSEMBLED.FITS
: The predicted probabilities of class 1 from the ensemble of phalanxes based on phalanxes.final.

BASE.CLASSIFIER.ARGS
: (Parsed) record of user-specified arguments for classifier.

PERFORMANCE.ARGS
: (Parsed) record of user-specified arguments for performance.

X
: User-provided data frame of explanatory variables.

Y
: User-provided binary response vector.

References

Tomal, J. H., Welch, W. J., & Zamar, R. H. (2015). Ensembling classification models based on phalanxes of variables with applications in drug discovery. *The Annals of Applied Statistics*, *9*(1), 69-93. doi: 10.1214/14AOAS778

See Also

summary.epx prints a summary of the results, and cv.epx assesses performance via cross-validation.

Examples

```
# Example with data(harvest)

## Phalanx-formation using a base classifier with 50 trees (default = 500)

set.seed(761)
model <- epx(x = harvest[, -4], y = harvest[, 4],
             classifier.args = list(ntree = 50))

## Phalanx-membership of explanatory variables at the four stages
## of phalanx formation (0 means not in a phalanx)
model$PHALANXES

## Summary of the final phalanxes (matches above)
summary(model)
## Not run:
## Parallel computing
clusters <- parallel::detectCores()
cl <- parallel::makeCluster(clusters)
doParallel::registerDoParallel(cl)
set.seed(761)
model.par <- epx(x = harvest[, -4], y = harvest[, 4],
```

```
                  computing = "parallel")
parallel::stopCluster(cl)

## End(Not run)
```

`harvest` *Simulated dataset for testing the EPX package*

Description

A simulated dataset from Yuan et al. (2012) with three explanatory variables. Demonstrates in a timely manner `epx`, the phalanx-formation algorithm in **EPX** and associated functions `summary.epx`, `predict.epx` , `plot.epx`, `cv.epx`, `hit.curve`.

Usage

```
harvest
```

Format

A dataframe with 190 rows and 4 variables:

LogP Octanol/water partition coefficient (-2 to 7).

MeltPt Melting point (120 to 280 degrees Celsius).

MolWt Molecular weight (200 to 800).

y The response variable where 1 denotes active and 0 inactive.

References

Yuan, Y., Chipman, H. A., & Welch, W. J. (2012). Harvesting Classification Trees for Drug Discovery. *Journal of Chemical Information and Modeling*, *52*(12), 3169-3180. doi: 10.1021/ci3000216

`hit.curve` *Plot hit curve*

Description

Plots the hit curve corresponding to `phat` and `y`.

Usage

```
hit.curve(y, phat, max.cutoff = min(100, length(y)), plot.hc = T, ...)
```

Arguments

y	True binary response vector where 1 denotes the relevant rare class.
phat	Vector of estimated probabilities of relevance.
max.cutoff	Maximum number of observations selected, equivalently the maximum shortlist cutoff; default is min(100, length(y)).
plot.hc	Whether to return a plot of the hit curve; default is TRUE.
...	Further arguments passed to or from other methods.

Details

Order the cases by decreasing phat (predicted probabilities of relevance) values, and plot the expected number and actual number of hits as cases are selected. Cases with tied phat values are grouped together. See plot.epx for plotting the hit curve for an "epx" object.

Value

Plot of the hit curve (if plot.hc = TRUE) and a list with the following vectors:

select	Number of observations in each tied phat group; select[1], select[2], ... are the numbers of observations with the largest predicted probability of relevance (max(phat)), the second largest value in phat, etc.
p	Unique phat values; p[1], p[2], ... are the largest value in phat, the second largest value in phat, etc.
nhits	Number of hits (truly relevant observations) in each tied phat group.
nhitlast	Number of hits after max.cutoff observations selected.

Examples

```
# Example with data(harvest)

## Phalanx-formation using a base classifier with 50 trees (default = 500)

set.seed(761)
model <- epx(x = harvest[, -4], y = harvest[, 4],
             classifier.args = list(ntree = 50))

## Plot hit curve for cross-validated predicted probabilities of relevence
set.seed(761)
model.cv <- cv.epx(model)
preds.cv <- model.cv[-nrow(model.cv), ncol(model.cv)]
cv.hc <- hit.curve(phat = as.numeric(preds.cv), y = model$Y)
```

IE *Calculate Initial Enhancement*

Description

Calculates initial enhancement (IE), which is the precision at one specific shortlist length (cutoff) normalised by the proportion of relevants in the total sample size (Tomal et al. 2015). Since IE is a rescaling of precision, we expect IE and AHR to lead to similar conclusions as an assessment metric for the EPX algorithm.

Usage

```
IE(y, phat, cutoff = length(y)/2, ...)
```

Arguments

`y`	True (binary) response vector where 1 is the rare/relevant class.
`phat`	Numeric vector of estimated probabilities of relevance.
`cutoff`	Shortlist cutoff length, and so must not exceed length of `y`; default is half the sample size.
`...`	Further arguments passed to or from other methods.

Details

Let c be the cutoff and $h(c)$ be the hitrate at c. Let also A be the total number of relevants and N be the total number of observations. IE is defined as

$$IE = h(c)/(A/N)$$

IE calculation does not change whether there are ties in `phat` or not.

Value

Numeric value of IE.

References

Tomal, J. H., Welch, W. J., & Zamar, R. H. (2015). Ensembling classification models based on phalanxes of variables with applications in drug discovery. *The Annals of Applied Statistics*, *9*(1), 69-93. doi: 10.1214/14AOAS778

Examples

```
## IE when there are no ties in phat:

resp <- c(1, 1, 0,  0,  0,  0,  0,   1,  0, 0)
prob <- (10:1) * 0.1
IE(y = resp, phat = prob, cutoff = 3)
```

```
# expect answer: (2/3) / (3/10)

## IE when there are ties
resp <- c(1, 1, 0,   0,   0,   0,   0,    1,   0, 0)
prob <- c(1, 1, 1, 0.4, 0.4, 0.3, 0.2, 0.15, 0.1, 0)
IE(y = resp, phat = prob, cutoff = 3)

# expect answer: same as above
```

`plot.epx` *Plot hit curve for an "*`epx`*" object*

Description

Plots the hit curve for the fitted values of an "epx" object.

Usage

```
## S3 method for class 'epx'
plot(x, max.cutoff = min(100, length(x$Y)), plot.hc = TRUE, ...)
```

Arguments

`x`	Object of class "epx".
`max.cutoff`	Maximum number of observations selected, equivalently the maximum shortlist cutoff; default is `min(100, length(x$Y))`.
`plot.hc`	Whether to make a plot of the hit curve; default is `TRUE`.
`...`	Further arguments passed to or from other methods.

Details

Order the cases by decreasing `phat` (predicted probabilities of relevance) values, and plot the expected number and actual number of hits as cases are selected. Cases with tied `phat` values are grouped together. See hit.curve in order to plot a hit curve in general.

Value

Plot of the hit curve (if `plot.hc = TRUE`) and a list with the following vectors:

`select`	Number of observations in each tied `phat` group; `select[1]`, `select[2]`, ... are the numbers of observations with the largest predicted probability of relevance (`max(phat)`), the second largest value in `phat`, etc.
`p`	Unique `phat` values; `p[1]`, `p[2]`, ... are the largest value in `phat`, the second largest value in `phat`, etc.
`nhits`	Number of hits (truly relevant observations) in each tied `phat` group.
`nhitlast`	Number of hits after `max.cutoff` observations selected.

Examples

```
# Example with data(harvest)

## Phalanx-formation using a base classifier with 50 trees (default = 500)

set.seed(761)
model <- epx(x = harvest[, -4], y = harvest[, 4],
             classifier.args = list(ntree = 50))

## Hit curve for model with default settings
model.hc <- plot(model)

## In the top 100 ranked observations selected, the number that are truly
## relevant is
model.hc$nhitlast

## Hit curve with max.cutoff at 150 (Note: Commented off for time.)
model.hc.150 <- plot(model, max.cutoff = 150)
model.hc.150$nhitlast  # Number of hits in top 150 ranked observations.
```

`predict.epx` *Predict with an "*`epx`*" object*

Description

Predicted values based on an "epx" object; may specify different base classifier arguments than those used for phalanx-formation.

Usage

```
## S3 method for class 'epx'
predict(object, newdata, classifier.args = list(), ...)
```

Arguments

`object`	Object of class "epx".
`newdata`	An optional data frame specifiying variables with which to predict; if omitted and `classifier.args` are not specified, the fitted (ensembled) values are used.
`classifier.args`	Additional arguments for the base classifier; same base classifier as that used for phalanx-formation (specified in epx).
`...`	Further arguments passed to or from other methods.

Value

Numeric vector of predicted values (double).

Examples

```
# Example with data(harvest)

## Phalanx-formation using a base classifier with 50 trees (default = 500)

set.seed(761)
model <- epx(x = harvest[, -4], y = harvest[, 4],
             classifier.args = list(ntree = 50))

## Predict training values without additional classifier.args and newdata
## returns the object's ENSEMBLED.FITS
all.equal(predict(model), model$ENSEMBLED.FITS)

## Predict training values using 100 trees (default = 500)
set.seed(761)
preds100 <- predict(model, classifier.args = list(ntree = 100))

## Predict test values by passing dataframe of test predictors to newdata as
## with the predict(model, newdata = . ) function etc.
```

RKL *Calculate rank last*

Description

The peformance measure rank last (RKL) is calculated as follows: after ranking the observations in decreasing order via phat, RKL is the rank of the last truly relevant observation. Hence, RKL can take on integer values from 1 to n, where n is the total number of observations. If there are ties, the last object in the tied group determines RKL. That is, if all n objects are tied at the first rank but only one object is truly relevant, RKL will have a value of n.

Usage

```
RKL(y, phat, ...)
```

Arguments

y	True (binary) response vector where 1 is the rare/relevant class.
phat	Numeric vector of estimated probabilities of relevance.
...	Further arguments passed to or from other methods.

Value

Numeric value of RKL.

Examples

```
## without ties in phat

resp <- c(rep(1, 50), rep(0, 50))
prob <- (1:100)*0.01
RKL(y = resp, phat = prob) # expect 100

resp <- c(rep(0, 50), rep(1, 50))
RKL(y = resp, phat = prob) # expect 50

## with ties in phat
resp <- sample(c(1, 0), 100, replace = TRUE)
prob <- rep(1, 100)
RKL(y = resp, phat = prob) # expect 100
```

`summary.epx` *Summarising an "*epx*" object*

Description

`summary` method for class "epx".

Usage

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

Arguments

`object`	Object of class "epx" returned by epx.
`...`	Further arguments passed to or from other methods.

Value

Prints a summary of the object returned by the phalanx-formation algorithm epx.

Examples

```
# Example with data(harvest)


## Phalanx-formation using a base classifier with 50 trees (default = 500)

set.seed(761)
model <- epx(x = harvest[, -4], y = harvest[, 4],
             classifier.args = list(ntree = 50))
summary(model)
```

```
## The summary agrees with
(model$PHALANXES)[[4]]
```

TOP1 *Calculate TOP1*

Description

The performance measure TOP1 is calculated as follows: after sorting the observations by their predicted probabilities of relevance (phat) in decreasing order so the first ranked observation has the highest probability of relevance, if the first ranked observation is truly relevant, TOP1 has a value of 1. Otherwise TOP1 is 0. If there are ties for the first rank, all the corresponding observations must be truly relevant for TOP1 to score 1.

Usage

```
TOP1(y, phat, ...)
```

Arguments

y	True (binary) response vector where 1 is the rare/relevant class.
phat	Numeric vector of estimated probabilities of relevance.
...	Further arguments passed to or from other methods.

Value

Numeric value of TOP1.

Examples

```
## with ties in phat

resp <- c(0, rep(1, 99))
prob <- rep(1, 100)
TOP1(y = resp, phat = prob)  # expect 0

resp <- c(1, 1, 1, rep(0, 95), 1, 1)
prob <- c(1, 1, 1, rep(0, 97))
TOP1(y = resp, phat = prob)  # expect 1

## no ties in phat
resp <- c(0, rep(1, 99))
prob <- (100:1)*0.01
TOP1(y = resp, phat = prob)  # expect 0

resp <- c(1, rep(0, 99))
TOP1(y = resp, phat = prob)  # expect 1
```

Index