Package: fbroc (via r-universe)

November 6, 2024

Type Package

Title Fast Algorithms to Bootstrap Receiver Operating Characteristics
Curves

Version 0.4.1

Date 2019-03-24

Description Implements a very fast C++ algorithm to quickly bootstrap
receiver operating characteristics (ROC) curves and derived
performance metrics, including the area under the curve (AUC)
and the partial area under the curve as well as the true and
false positive rate. The analysis of paired receiver operating
curves is supported as well, so that a comparison of two
predictors is possible. You can also plot the results and
calculate confidence intervals. On a typical desktop computer
the time needed for the calculation of 100000 bootstrap
replicates given 500 observations requires time on the order of
magnitude of one second.

License GPL-2

URL http://www.epeter-stats.de/roc-curve-analysis-with-fbroc/

BugReports http://github.com/erikpeter/fbroc/issues

Depends R (>= 3.2.0), ggplot2, methods, stats, utils

Imports Rcpp

LinkingTo Rcpp

LazyData true

RoxygenNote 6.1.1

Encoding UTF-8

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

RemoteUrl https://github.com/erikpeter/fbroc

RemoteRef HEAD

RemoteSha 14dbfa1750ab4093b64228316c002d357656786e

Contents

---

boot.paired.roc	*Bootstrap paired ROC curves*

Description

Given two numerical predictors for the same outcome on the same set of samples, this functions enables the bootstrapping of the paired ROC curves of the two prediction models. While bootstrapping the same set of samples are used for both curves in each iteration, preserving the correlation between the two models.

Usage

```
boot.paired.roc(pred1, pred2, true.class, stratify = TRUE,
  n.boot = 1000, use.cache = FALSE, tie.strategy = NULL)
```

Arguments

pred1	Numerical predictions for the first classifier.
pred2	Numerical predictions for the second classifier.
true.class	A logical vector. TRUE indicates the sample belonging to the positive class.
stratify	Logical. Indicates whether we use stratified bootstrap. Default to TRUE. Non-stratified bootstrap is not yet implemented.

n.boot	A number that will be coerced to integer. Specified the number of bootstrap replicates. Defaults to 1000.
use.cache	If true the bootstrapping results for the ROC curve will be pre-cached. This increases speed when the object is used often, but also takes up more memory.
tie.strategy	How to handle ties. See details below.

Value

A list of class fbroc.paired.roc, containing the elements:

prediction1	Input predictions for first model.
prediction2	Input predictions for second model.
true.class	Input classes.
n.thresholds1	Number of thresholds of the first predictor.
n.thresholds2	Number of thresholds of the second predictor.
n.boot	Number of bootstrap replicates.
use.cache	Indicates if cache is used for this ROC object.
tie.strategy	Used setting how to handle ties in predictors.
n.pos	Number of positive observations.
n.neg	Number of negative observations.
roc1	A data.frame containing the thresholds of the first ROC curve and the TPR and FPR at these thresholds.
roc2	A data.frame containing the thresholds of the second ROC curve and the TPR and FPR at these thresholds.
auc1	The AUC of the first ROC curve.
auc2	The AUC of the second ROC curve.
boot.tpr1	If the cache is enabled, a matrix containing the bootstrapped TPR at the thresholds for the first predictor.
boot.fpr1	If the cache is enabled, a matrix containing the bootstrapped FPR at the thresholds for the first predictor.
boot.tpr2	If the cache is enabled, a matrix containing the bootstrapped TPR at the thresholds for the second predictor.
boot.fpr2	If the cache is enabled, a matrix containing the bootstrapped FPR at the thresholds for the second predictor.

Caching

If you enable caching, boot.roc calculates the requested number of bootstrap samples and saves the TPR and FPR values for each iteration. This can take up a sizable portion of memory, but it speeds up subsequent operations. This can be useful if you plan to use the ROC curve multiple fbroc functions.

Ties

You can set this parameter to either 1 or 2. If your numerical predictor has no ties, both settings will produce the same results. If you set tie.strategy to 1 the ROC curve is built by connecting the TPR/FPR pairs for neighboring thresholds. A tie.strategy of 2 indicates that the TPR calculated at a specific FPR is the best TPR at a FPR smaller than or equal than the FPR specified. Defaults to 2.

See Also

boot.roc, plot.fbroc.paired.roc, perf.fbroc.paired.roc

Examples

```
data(roc.examples)
# Do not use cache
example <- boot.paired.roc(roc.examples$Cont.Pred, roc.examples$Cont.Pred.Outlier,
                           roc.examples$True.Class, n.boot = 500)
perf(example, "auc") # estimate difference in auc
perf(example, "tpr", fpr = 0.5) # estimate difference in TPR at a FPR of 50%
plot(example) # show plot
# Cached mode
example <- boot.paired.roc(roc.examples$Cont.Pred, roc.examples$Cont.Pred.Outlier,
                           roc.examples$True.Class, n.boot = 1000, use.cache = TRUE)
conf(example, conf.for = "tpr", steps = 10) # get confidence regions for TPR at FPR
conf(example, conf.for = "fpr", steps = 10) # get confidence regions for FPR at TPR
perf(example, "fpr", tpr = 0.9) # estimate difference in FPR at a TPR of 90%
```

---

boot.roc   *Bootstrap ROC curve*

Description

boot.roc calculates the ROC curve, initializes the settings and calculates the bootstrap results for the true and false positive rate at every relevant threshold. Missing values are removed with a warning prior to bootstrapping.

Usage

```
boot.roc(pred, true.class, stratify = TRUE, n.boot = 1000,
  use.cache = FALSE, tie.strategy = NULL)
```

Arguments

pred	A numeric vector. Contains predictions. boot.roc assumes that a high prediction is evidence for the observation belonging to the positive class.
true.class	A logical vector. TRUE indicates the sample belonging to the positive class.
stratify	Logical. Indicates whether we use stratified bootstrap. Default to TRUE. Non-stratified bootstrap is not yet implemented.

n.boot	A number that will be coerced to integer. Specified the number of bootstrap replicates. Defaults to 1000.
use.cache	If true the bootstrapping results for the ROC curve will be pre-cached. This increases speed when the object is used often, but also takes up more memory.
tie.strategy	How to handle ties. See details below.

Value

A list of class fbroc.roc, containing the elements:

prediction	Input predictions.
true.class	Input classes.
roc	A data.frame containing the thresholds of the ROC curve and the TPR and FPR at these thresholds.
n.thresholds	Number of thresholds.
n.boot	Number of bootstrap replicates.
use.cache	Indicates if cache is used for this ROC object
tie.strategy	Used setting how to handle ties in predictors.
n.pos	Number of positive observations.
n.neg	Number of negative observations.
auc	The AUC of the original ROC curve.
boot.tpr	If the cache is enabled, a matrix containing the bootstrapped TPR at the thresholds.
boot.fpr	If the cache is enabled, a matrix containing the bootstrapped FPR at the thresholds.

Caching

If you enable caching, boot.roc calculates the requested number of bootstrap samples and saves the TPR and FPR values for each iteration. This can take up a sizable portion of memory, but it speeds up subsequent operations. This can be useful if you plan to use the ROC curve multiple fbroc functions.

Ties

You can set this parameter to either 1 or 2. If your numerical predictor has no ties, both settings will produce the same results. If you set tie.strategy to 1 the ROC curve is built by connecting the TPR/FPR pairs for neighboring thresholds. A tie.strategy of 2 indicates that the TPR calculated at a specific FPR is the best TPR at a FPR smaller than or equal than the FPR specified. Defaults to 2.

See Also

http://www.epeter-stats.de/roc-curves-and-ties/, plot.fbroc.roc, print.fbroc.roc

Examples

```
y <- rep(c(TRUE, FALSE), each = 500)
x <- rnorm(1000) + y
result.boot <- boot.roc(x, y)
```

---

boot.tpr.at.fpr   *Process bootstrapped TPR/FPR at thresholds matrix into TPR at FPR matrix*

Description

Usually `fbroc` calculates the TPR and FPR at the thresholds of the ROC curve. per bootstrap replicate. This can be calculated quickly, but is often not convenient to work with. Therefore `boot.tpr.at.fpr` instead gets the TPR along a sequence of different values for the FPR.

Usage

```
boot.tpr.at.fpr(roc, steps = roc$n.neg)
```

Arguments

roc	Object of class `fbroc.roc`.
steps	Number of discrete steps for the FPR at which the TPR is calculated. TPR confidence intervals are given for all FPRs in `seq(0, 1, by = (1 / steps))`. Defaults to `n.neg`, thus covering all possible values.

Value

Matrix containing the TPR bootstrap replicates for the discrete FPR steps.

See Also

[boot.roc](#)

---

conf   *Generic S3 function to calculate confidence regions for ROC curves*

Description

For using this on individual ROC curves as implemented by objects of class fbroc.roc see [conf.fbroc.roc](#). For paired ROC curves (class conf.paired.roc) see [conf.fbroc.paired.roc](#).

Usage

```
conf(roc, ...)
```

Arguments

roc	The object for which to calculate the performance.
...	Further arguments to perf.

See Also

[conf.fbroc.roc](), [conf.fbroc.paired.roc]()

---

conf.fbroc.paired.roc   *Generates confidence intervals for the difference in TPR between two predictors for a range of FPRs or vice versa*

Description

Calculates confidence intervals for the difference in TPR at different FPR values or vice versa. The stepsize at which the TPR or FPR is calculated can be set as needed.

Usage

```
## S3 method for class 'fbroc.paired.roc'
conf(roc, conf.level = 0.95,
  conf.for = "TPR", steps = 250, ...)
```

Arguments

roc	An object of class fbroc.paired.roc.
conf.level	Confidence level to be used for the confidence intervals. Defaults to 0.95.
conf.for	Use "tpr" to get confidence regions for the TPR at specific FPRs. Use "fpr" instead for confidence regions for the FPR at specific TPRs.
steps	Number of discrete steps at which the requested rate and the confidence region is calculated. Defaults to 250.
...	Further arguments, that are not used at this time.

Value

A data.frame containing either discrete TPR steps and estimates and confidence bounds for the difference FPR or vice versa, depending upon conf.for.

Details

This function only gives estimates and confidence for the difference in the requested rate (either True Positive Rate or False Positive Rate) between the first and the second classifier. The values given are positive iff the first classifier has a higher rate. To get confidence regions for either of the two underlying ROC curves use conf on the result of extract.roc.

See Also

[boot.paired.roc](), [conf.fbroc.roc](),[extract.roc]()

Examples

```
data(roc.examples)
example <- boot.paired.roc(roc.examples$Cont.Pred, roc.examples$Cont.Pred.Outlier,
                          roc.examples$True.Class, n.boot = 100)
conf(example, conf.for = "tpr", steps = 10) # get confidence regions for Delta TPR at FPR
conf(example, conf.for = "fpr", steps = 10) # get confidence regions for Delta FPR at TPR
```

---

conf.fbroc.roc   *Generates confidence intervals for the TPR for a range of FPRs or*
                              vice versa

Description

Calculates confidence intervals for the TPR at different FPR values or vice versa. The stepsize at which the TPR or FPR is calculated can be set as needed.

Usage

```
## S3 method for class 'fbroc.roc'
conf(roc, conf.level = 0.95, conf.for = "tpr",
  steps = 250, ...)
```

Arguments

roc	Object of class fbroc.roc.
conf.level	Confidence level to be used for the confidence intervals. Defaults to 0.95.
conf.for	Use "tpr" to get confidence regions for the TPR at specific FPRs. Use "fpr" instead for confidence regions for the FPR at specific TPRs.
steps	Number of discrete steps at which the requested rate and the confidence region is calculated. Defaults to 250.
...	Further arguments, that are not used at this time.

Value

A data.frame containing either discrete TPR steps and estimates and confidence bounds for FPR or vice versa, depending upon conf.for.

See Also

[boot.roc]()

Examples

```
data(roc.examples)
example <- boot.roc(roc.examples$Cont.Pred, roc.examples$True.Class,
                    n.boot = 100)
conf(example, conf.for = "tpr", steps = 10) # get confidence regions for TPR at FPR
conf(example, conf.for = "fpr", steps = 10) # get confidence regions for FPR at TPR
```

---

extract.roc	*Extracts one from two paired ROC curves from a* fbroc.paired.roc *object*

Description

Given paired ROC curves it can be helpful to look at them in isolation as well. This functions allows the extraction of one of the paired ROC curves as a fbroc.roc object without recalculating the ROC curve.

Usage

```
extract.roc(x, index)
```

Arguments

x	Object of class fbroc.paired.roc.
index	A number specifying which of the two ROC curves should be returned. Needs to be 1 or 2.

Value

An object of class fbroc.roc containing all data about the requested ROC curve

Note

Due to the way the predictions are reordered internally, using use.cache to save the bootstrap results for paired ROC curves and then extracting one of the two curves will not yield the same values as when the ROC curve is bootstrapped as a single curve using fbroc.roc.

See Also

boot.paired.roc

Examples

```
data(roc.examples)
example <- boot.paired.roc(roc.examples$Cont.Pred, roc.examples$Cont.Pred.Outlier,
                           roc.examples$True.Class, n.boot = 100)
roc1 <- extract.roc(example, 1)
roc1.equivalent <- boot.roc(roc.examples$Cont.Pred, roc.examples$True.Class,
                            n.boot = 100)
print(identical(roc1, roc1.equivalent)) # roc1 and roc1.equivalent will be the same
# This does not hold when use.cache = TRUE. See the note above.
```

---

fbroc   *fbroc: A package for fast bootstrap analysis and comparison of ROC*
                                curves

Description

Fbroc enables the fast bootstrap analysis and comparison of ROC curves for simulation studies
and shiny applications by using a fast algorithm where the cost of a single bootstrap replicate is
$O(n)$, with n denoting the number of observations. The algorithm is implemented in C++ to further
increase the efficiency. On a typical desktop computer the time needed for the calculation of 100000
bootstrap replicates given 500 observations requires time on the order of magnitude of one second.
The ROC curve as used shows the True Positive Rate (TPR) as a function of the False Positive
Rate (FPR). The package also support the analysis of paired ROC curves, where we compare two
predictions given for the same set of samples.

Important fbroc functions

[boot.roc](#) Use boot.roc to bootstrap a ROC curve.

[boot.paired.roc](#) Use boot.paired.roc to bootstrap two paired ROC curves.

[conf](#) Calculate confidence regions for the ROC curve.

[perf](#) Estimate performance and calculate confidence intervals.

Example Data

fbroc also contains the example data set [roc.examples](#), which you can use to test the functionality
of the package. This data set contains simulated data and not an real application.

Details

The algorithm works by first determining the critical thresholds of the ROC curve - cutoffs at which
the curve changes directions. Each observation is then linked to the specific thresholds at which
they first cause a change in the TPR or FPR. Calculating this link and directly bootstrapping that
link allows us to construct the bootstrapped ROC curve very quickly. Since multiple observation
can be linked to the same threshold, it is difficult to implement the algorithm efficiently in R. This
is why fbroc implements it in C++.

When bootstrapping paired ROC curves, the packages takes care of using the same set of samples for both predictors in each iteration of the bootstrap. This preserves the correlation structure between both predictors.

All bootstrap confidence interval are based on the percentile method.

Notes

Package `fbroc` is still in an early development stage. Currently it supports bootstrapping the confidence region of single and paired ROC curves, as well as the AUC, partial AUC, the FPR at a fixed TPR and vice versa. More sophisticated bootstrap confidence interval calculation and improved documentation will be added at a later time.

References

Efron, B., & Tibshirani, R. (1998). *An introduction to the bootstrap.* Boca Raton, Fla: Chapman & Hall/CRC.

Donna Katzman McClish. (1989). *Analyzing a Portion of the ROC Curve.* Medical Decision Making, <http://mdm.sagepub.com/content/9/3/190.abstract>.

Examples

```
data(roc.examples)
# work with a single ROC curves
result.boot <- boot.roc(roc.examples$Cont.Pred, roc.examples$True.Class, n.boot = 100)
plot(result.boot)
perf(result.boot, "auc")
perf(result.boot, "auc", conf.level = 0.99)
perf(result.boot, "tpr", conf.level = 0.95, fpr = 0.1)
conf(result.boot, steps = 10)
# work with paired ROC curves
result.boot <- boot.paired.roc(roc.examples$Cont.Pred, roc.examples$Cont.Pred.Outlier,
                               roc.examples$True.Class, n.boot = 100)
plot(result.boot)
perf(result.boot, "auc")
perf(result.boot, "auc", conf.level = 0.99)
perf(result.boot, "tpr", conf.level = 0.95, fpr = 0.1)
conf(result.boot, steps = 10)
```

---

perf	*Generic S3 function to calculate performance estimates for ROC curves*

Description

For using this on individual ROC curves as implemented by objects of class fbroc.roc see `perf.fbroc.roc`. For paired ROC curves (class fbroc.paired.roc) see `perf.fbroc.paired.roc`.

Usage

```
perf(roc, ...)
```

Arguments

roc	The object for which to calculate the performance.
...	Further arguments to perf.

See Also

perf.fbroc.roc, perf.fbroc.paired.roc

---

perf.fbroc.paired.roc   *Calculate performance for paired bootstrapped ROC curves*

Description

For a given metric this calculates the difference in performance between two paired predictors stored in an object of class fbroc.paired.roc in addition to their individual performance.

Usage

```
## S3 method for class 'fbroc.paired.roc'
perf(roc, metric = "auc", conf.level = 0.95,
  tpr = NULL, fpr = NULL, correct.partial.auc = TRUE,
  show.partial.auc.warning = TRUE, ...)
```

Arguments

roc	An object of class fbroc.paired.roc.
metric	A performance metric. Select "auc" for the AUC, "partial.auc" for the partial AUC, "tpr" for the TPR at a fixed FPR and "fpr" for the FPR at a fixed TPR.
conf.level	The confidence level of the confidence interval.
tpr	The fixed TPR at which the FPR is to be evaluated when fpr is selected as metric. If partial AUC is investigated, then an TPR interval over which the partial area is to be calculated.
fpr	The fixed FPR at which the TPR is to be evaluated when tpr is selected as metric. If partial AUC is investigated, then an FPR interval over which the partial area is to be calculated.
correct.partial.auc	
	Corrects partial AUC for easier interpretation using McClish correction. Details are given below. Defaults to TRUE.
show.partial.auc.warning	
	Whether to give warnings for partial AUCs below 0.5. Defaults to true.
...	Further arguments, that are not used at this time.

Note on partial AUC correction

The partial AUC is hard to interpret without considering the range on which it is calculated. Not only does the partial AUC scale with the width of the interval over which it is calculated, but it also depends on where the interval is located. For example, if the ROC Curve is integrated over the FPR interval [0, 0.1] a completely random and non-discrimate classifier would have a partial AUC of 0.05, but the same ROC curve integrated over the interval [0.9, 1] would yield a partial AUC of 0.95.

The correction by McClish produces a corrected partial AUC given by:

$$\frac{1}{2}\left(1 + \frac{\text{partialAUC} - \text{auc.min}}{\text{auc.max} - \text{auc.min}}\right)$$

Here auc.min is the AUC achieved by the non-discriminate classifier and auc.max is the AUC achieved by a perfect classifier. Thus, a non-discriminative classifier will always have an AUC of 0.5 and a perfect one classifier will always have a partial AUCs of 1.

Unfortunately, the corrected partial AUC cannot be interpreted in a meaningful way if the curve is below the non-discriminate classifier, producing corrected partial AUCs values below 0.5. For this reason, fbroc will give a warning if the bootstrap produces corrected partial AUC values below 0.5.

References

Donna Katzman McClish. (1989). *Analyzing a Portion of the ROC Curve*. Medical Decision Making, <http://mdm.sagepub.com/content/9/3/190.abstract>.

Examples

```
data(roc.examples)
example <- boot.paired.roc(roc.examples$Cont.Pred, roc.examples$Cont.Pred.Outlier,
                           roc.examples$True.Class, n.boot = 100)
perf(example, metric = "auc")
# Get difference in TPR at a FPR of 20%
perf(example, metric = "tpr", fpr = 0.2)
perf(example, metric = "partial.auc", fpr = c(0, 0.25),
     show.partial.auc.warning = FALSE)
```

---

perf.fbroc.roc	*Calculate performance for bootstrapped ROC curve*

Description

Calculates different performance metric for ROC curves based on the bootstrap results saved in an object of class fbroc.roc. Confidence intervals are included.

Usage

```
## S3 method for class 'fbroc.roc'
perf(roc, metric = "auc", conf.level = 0.95,
  tpr = NULL, fpr = NULL, correct.partial.auc = TRUE,
  show.partial.auc.warning = TRUE, ...)
```

Arguments

`roc`	An object of class `fbroc.roc`.
`metric`	A performance metric. Select "auc" for the AUC, "partial.auc" for the partial AUC, "tpr" for the TPR at a fixed FPR and "fpr" for the FPR at a fixed TPR.
`conf.level`	The confidence level of the confidence interval.
`tpr`	The fixed TPR at which the FPR is to be evaluated when `fpr` is selected as metric. If partial AUC is investigated, then an TPR interval over which the partial area is to be calculated.
`fpr`	The fixed FPR at which the TPR is to be evaluated when `tpr` is selected as metric. If partial AUC is investigated, then an FPR interval over which the partial area is to be calculated.
`correct.partial.auc`	
	Corrects partial AUC for easier interpretation using McClish correction. Details are given below. Defaults to TRUE.
`show.partial.auc.warning`	
	Whether to give warnings for partial AUCs below 0.5. Defaults to true.
`...`	Further arguments, that are not used at this time.

Value

A list of class `fbroc.perf`, containing the elements:

`Observed.Performance`	
	The observed performance.
`CI.Performance`	Quantile based confidence interval for the performance.
`conf.level`	Confidence level of the confidence interval.
`metric`	Used performance metric.
`params`	Parameters used to further specifiy metric, e.g. fixed TPR.
`n.boot`	Number of bootstrap replicates used.
`boot.results`	Performance in each bootstrap replicate.

Note on partial AUC correction

The partial AUC is hard to interpret without considering the range on which it is calculated. Not only does the partial AUC scale with the width of the interval over which it is calculated, but it also depends on where the interval is located. For example, if the ROC Curve is integrated over the FPR interval [0, 0.1] a completely random and non-discrimate classifier would have a partial AUC of 0.05, but the same ROC curve integrated over the interval [0.9, 1] would yield a partial AUC of 0.95.

The correction by McClish produces a corrected partial AUC given by:

$$\frac{1}{2}\Big(1 + \frac{\text{partialAUC} - \text{auc.min}}{\text{auc.max} - \text{auc.min}}\Big)$$

Here auc.min is the AUC achieved by the non-discriminate classifier and auc.max is the AUC achieved by a perfect classifier. Thus, a non-discriminative classifier will always have an AUC of 0.5 and a perfect one classifier will always have a partial AUCs of 1.

Unfortunately, the corrected partial AUC cannot be interpreted in a meaningful way if the curve is below the non-discriminate classifier, producing corrected partial AUCs values below 0.5. For this reason, fbroc will give a warning if the bootstrap produces corrected partial AUC values below 0.5.

References

Donna Katzman McClish. (1989). *Analyzing a Portion of the ROC Curve.* Medical Decision Making, <http://mdm.sagepub.com/content/9/3/190.abstract>.

See Also

`boot.roc`, `print.fbroc.perf`, `plot.fbroc.perf`

Examples

```
y <- rep(c(TRUE, FALSE), each = 100)
x <- rnorm(200) + y
result.boot <- boot.roc(x, y, n.boot = 100)
perf(result.boot, "auc")
perf(result.boot, "auc", conf.level = 0.99)
perf(result.boot, "partial.auc", fpr = c(0, 0.25), show.partial.auc.warning = FALSE)
```

---

plot.fbroc.conf   *Plots function for object of classfbroc.conf*

Description

Given an object of class `fbroc.conf` this function plots the contained estimates for the confidence region of the ROC curve.

Usage

```
## S3 method for class 'fbroc.conf'
plot(x, col = "blue", fill = "royalblue1",
  print.plot = TRUE, ...)
```

Arguments

x	Object of class `fbroc.conf` to be plotted.
col	Color of the curve to be drawn.
fill	Fill of the confidence region.
print.plot	Logical specifying whether the plot should be printed.
...	Further arguments passed to or from other methods.

Value

A ggplot, so that the user can customize the plot further.

See Also

conf.fbroc.roc

Examples

```
data(roc.examples)
example <- boot.roc(roc.examples$Cont.Pred, roc.examples$True.Class, n.boot = 100)
# Confidence regions for TPR at specific FPR values
tpr.conf <- conf(example, conf.for = "tpr", steps = 50)
plot(tpr.conf)
# Confidence regions for FPR at specific TPR values
fpr.conf <- conf(example, conf.for = "fpr", steps = 50)
plot(fpr.conf)
```

```
plot.fbroc.conf.paired
```

                        Plots function for object of class `fbroc.conf.paired`

Description

Given an object of class `fbroc.conf.paired` this function plots the contained estimates for the confidence region of the *difference* between the two individual ROC curves.

Usage

```
## S3 method for class 'fbroc.conf.paired'
plot(x, col = "blue", fill = "royalblue1",
  print.plot = TRUE, ...)
```

Arguments

x	Object of class `fbroc.conf.paired` to be plotted.
col	Color of the curve to be drawn.
fill	Fill of the confidence region.
print.plot	Logical specifying whether the plot should be printed.
...	Further arguments passed to or from other methods.

Value

A ggplot, so that the user can customize the plot further.

See Also

conf.fbroc.paired.roc

Examples

```
data(roc.examples)
example <- boot.paired.roc(roc.examples$Cont.Pred, roc.examples$Cont.Pred.Outlier,
                           roc.examples$True.Class, n.boot = 1000)
# Confidence regions for the difference in TPR at specific FPR values
tpr.conf <- conf(example, conf.for = "tpr", steps = 50)
plot(tpr.conf)
# Confidence regions for the difference in FPR at specific TPR values
fpr.conf <- conf(example, conf.for = "fpr", steps = 50)
plot(fpr.conf)
```

---

plot.fbroc.paired.roc   *Plots a* fbroc.paired.roc *object*

Description

Plots a fbroc.paired.roc object and shows the two paired ROC curves. The confidence regions
for the ROC curves and the performance estimates and confidence bounds for a specified metric can
also be included in the plot.

Usage

```
## S3 method for class 'fbroc.paired.roc'
plot(x, col1 = "blue", fill1 = "dodgerblue",
  col2 = "darkgreen", fill2 = "seagreen1", print.plot = TRUE,
  show.conf = TRUE, conf.level = 0.95, steps = 250,
  show.metric = NULL, show.area = !show.conf, text.size.perf = 6,
  ...)
```

Arguments

x	An object of class fbroc.paired.roc.
col1	Color in which the ROC curve of the first classifier is drawn.
fill1	Color used for areas (confidence regions, AUCs and partial AUCs) belonging to the first ROC curve.
col2	Color in which the ROC curve of the second classifier is drawn.
fill2	Color used for areas (confidence regions, AUCs and partial AUCs) belonging to the second ROC curve.
print.plot	Logical specifying whether the plot should be printed.
show.conf	Logical specifying whether the confidence region should be plotted.
conf.level	Confidence level of the confidence region.
steps	Number of discrete steps for the FPR at which the TPR is calculated. TPR confidence intervals are given for all FPRs in seq(0, 1, by = (1 / steps)). Defaults to 250.

show.metric	Character specifying which metric to display. See [perf.fbroc.roc](#) for details. Defaults to NULL, which means that no metric is displayed.
show.area	Whether to shade the AUC or partial AUC area. Defaults to !show.conf.
text.size.perf	Size of the text display when show.metric is set to TRUE. Defaults to 6.
...	further arguments passed to [perf.fbroc.paired.roc](#).

Value

A ggplot, so that the user can customize the plot further.

See Also

[boot.paired.roc](#)

Examples

```
data(roc.examples)
example <- boot.paired.roc(roc.examples$Cont.Pred, roc.examples$Cont.Pred.Outlier,
                           roc.examples$True.Class, n.boot = 100)
plot(example) # standard plot, no metric shown
plot(example, show.metric = "auc") # Include information about the AUC
plot(example, show.metric = "auc", show.conf = FALSE) # Show area instead
# Highlight TPR at an FPR of 20%
plot(example, show.metric = "tpr", fpr = 0.2)
plot(example, show.metric = "partial.auc", fpr = c(0.2, 0.4),
     show.conf = FALSE, show.partial.auc.warning = FALSE) # Show area
```

---

plot.fbroc.perf	*Plots ROC based performance metric as histogram*

Description

Given an object of class fbroc.perf this function plots the results of the bootstrap as a histogram. The confidence interval is also included by default.

Usage

```
## S3 method for class 'fbroc.perf'
plot(x, bins = NULL, col = "white",
  fill = "lightblue", print.plot = TRUE, show.conf = TRUE,
  conf.text = TRUE, ...)
```

Arguments

x	Object of class `fbroc.perf` to be plotted.
bins	Number of bins for histogram. Default value depends on the number of bootstrap values and the number of unique bootstrap performance values.
col	Color of outline of histogram bars. Defaults to white.
fill	Fill of histogram bars. Defaults to lightblue.
print.plot	Logical specifying whether the plot should be printed.
show.conf	Logical specifying whether the confidence interval should be displayed.
conf.text	Logical specifying whether the confidence interval limits should also be displayed as text.
...	Further arguments passed to or from other methods.

Value

A ggplot, so that the user can customize the plot further.

See Also

[perf.fbroc.roc](#)

Examples

```
y <- rep(c(TRUE, FALSE), each = 500)
x <- rnorm(1000) + y
result.boot <- boot.roc(x, y, n.boot = 1000)
result.perf <- perf(result.boot, "auc")
plot(result.perf)
```

---

plot.fbroc.perf.paired

Plots the difference between the bootstrapped performance estimate of the first and the second classifier.

Description

Given an object of class `fbroc.perf.paired` this function plots the difference between the bootstrapped performance estimate of the first and the second classifier as a histogram. the bootstrap as an histogram. The confidence interval is also shown by default.

Usage

```
## S3 method for class 'fbroc.perf.paired'
plot(x, bins = NULL, col = "white",
  fill = "lightblue", print.plot = TRUE, show.conf = TRUE,
  conf.text = TRUE, ...)
```

Arguments

x	An object of class `fbroc.perf.paired`.
bins	Number of bins for histogram. Default value depends on the number of bootstrap values and the number of unique bootstrap performance values.
col	Color of outline of histogram bars. Defaults to white.
fill	Fill of histogram bars. Defaults to lightblue.
print.plot	Logical specifying whether the plot should be printed.
show.conf	Logical specifying whether the confidence interval should be displayed.
conf.text	Logical specifying whether the confidence interval limits should also be displayed as text.
...	Further arguments passed to or from other methods.

Value

A ggplot, so that the user can customize the plot further.

See Also

[`perf.fbroc.paired.roc`](#)

Examples

```
data(roc.examples)
example <- boot.paired.roc(roc.examples$Cont.Pred, roc.examples$Cont.Pred.Outlier,
                          roc.examples$True.Class, n.boot = 100)
auc.diff <- perf(example, ”auc”)
plot(auc.diff)
```

---

plot.fbroc.roc   *Plots a* fbroc.roc *object*

Description

Plot a `fbroc.roc` object and shows the ROC curve. The confidence region for the ROC curve and the result for a specified performance metric can also be included in the plot.

Usage

```
## S3 method for class 'fbroc.roc'
plot(x, col = ”blue”, fill = ”royalblue1”,
  print.plot = TRUE, show.conf = TRUE, steps = 250,
  conf.level = 0.95, show.metric = NULL, text.size.perf = 6,
  show.area = !show.conf, ...)
```

Arguments

x	Object of class `fbroc.roc`.
col	Color used for the curve. Defaults to blue.
fill	Color used for areas (confidence regions, AUCs and partial AUCs).
print.plot	Logical specifying whether the plot should be printed.
show.conf	Logical specifying whether the confidence region should be plotted.
steps	Number of discrete steps for the FPR at which the TPR is calculated. TPR confidence intervals are given for all FPRs in `seq(0, 1, by = (1 / steps))`. Defaults to 250.
conf.level	Confidence level of the confidence region.
show.metric	Character specifying which metric to display. See `perf.fbroc.roc` for details. Defaults to NULL, which means that no metric is displayed.
text.size.perf	Size of the text display when show.metric is set to TRUE. Defaults to 6.
show.area	Whether to shade the AUC or partial AUC area. Defaults to !show.conf.
...	further arguments passed to `perf.fbroc.roc`.

Value

A ggplot, so that the user can customize the plot further.

See Also

`boot.roc`, `perf.fbroc.roc`

Examples

```
y <- rep(c(TRUE, FALSE), each = 100)
x <- rnorm(200) + y
result.boot <- boot.roc(x, y, n.boot = 100)
plot(result.boot)
plot(result.boot, show.metric = "auc")
plot(result.boot, show.metric = "auc", show.conf = FALSE) # show area instead
plot(result.boot, show.metric = "tpr", fpr = 0.2)
plot(result.boot, show.metric = "partial.auc", fpr = c(0, 0.5),
     show.partial.auc.warning = FALSE)
plot(result.boot, show.metric = "partial.auc", fpr = c(0, 0.5), show.conf = FALSE,
     show.partial.auc.warning = FALSE)  # show area instead
```

---

print.fbroc.perf   *Prints information about a* fbroc.perf *object*

Description

Prints the information about the bootstrap results for an object of class fbroc.perf. This information includes the number of bootstrap replicates, the metric used and the estimate with confidence interval.

Usage

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

Arguments

x   Object of class fbroc.perf.

...   further arguments passed to or from other methods.

Value

Character containing the text that is also printed.

See Also

[perf.fbroc.roc](#)

---

print.fbroc.perf.paired

Prints information about a fbroc.perf.paired *object*

Description

Prints the information about the bootstrap results for an object of class fbroc.perf.paired. This information includes the number of bootstrap replicates, the metric used and estimates for both the individual classifiers and the difference in performance including confidence intervals. Finally, an estimate for the correlation between the performance estimates of the two classifiers is also given.

Usage

```
## S3 method for class 'fbroc.perf.paired'
print(x, ...)
```

Arguments

x	Object of class `fbroc.perf`.
...	further arguments passed to or from other methods.

Value

Character containing the text that is also printed.

See Also

[perf.fbroc.paired.roc](#)

---

print.fbroc.roc   *Prints information about a* fbroc.roc *object*

Description

Prints the information about the bootstrap results for an object of class `fbroc.roc`. This information includes the number of bootstrap replicates, the time spent on bootstrapping, the AUC and the memory usage of the object.

Usage

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

Arguments

x	Object of class `fbroc.roc`.
...	further arguments passed to or from other methods.

Value

Character containing the text that is also printed.

See Also

[boot.roc](#)

roc.examples	*Examples of predictions for ROC curve construction*

Description

Contains simulated data that can be used as examples for generating ROC curves. Both a continuous and a discrete predictor are included. For both cases there is a version with outliers and one without.

Usage

```
roc.examples
```

Format

A data.frame with 160 rows and 5 variables:

True.Class True class label of the observation

Cont.Pred Predictions for which the binormal model for ROC curves holds. Predictions for both the positive and negative class follows a normal distribution with unit standard deviation and means 2 and 0 respectively.

Cont.Pred.Outlier Same as above, with some extreme outliers in the negative class.

Disc.Pred Example of a discrete predictor. Predictions for the negative class are integer values between 1 and 8, positive samples have integer predictions between 7 and 14.

Disc.Pred.Outlier Same as above, with some extreme outliers in the negative class.

Index