Package 'multiROC'

June 26, 2018

Title Calculating and Visualizing ROC and PR Curves Across Multi-Class Classifications

Version 1.1.1

Description
Tools to solve real-world problems with multiple classes classifications by computing the areas under ROC and PR curve via micro-averaging and macro-averaging. The vignettes of this package can be found via <https://github.com/WandeRum/multiROC>. The methodology is described in V. Van Asch (2013) <https://www.clips.uantwerpen.be/~vincent/pdf/microaverage.pdf> and Pedregosa et al. (2011) <http://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html>.

License GPL-3

Encoding UTF-8

LazyData true

Imports zoo, magrittr, boot, stats

Suggests dplyr, ggplot2

NeedsCompilation no

Repository CRAN

Date/Publication 2018-06-26 20:24:05 UTC

RoxygenNote 6.0.1.9000

Author Runmin Wei [aut, cre],
Jingye Wang [aut],
Wei Jia [ctb]

Maintainer Runmin Wei <runmin@hawaii.edu>

R topics documented:

`cal_auc` *Area under ROC curve*

Description

This function calculates the area under ROC curve

Usage

```
cal_auc(X, Y)
```

Arguments

`X`	A vector of true positive rate
`Y`	A vector of false positive rate, same length with TPR

Details

This function calculates the area under ROC curve.

Value

A numeric value of AUC will be returned.

References

https://www.r-bloggers.com/calculating-auc-the-area-under-a-roc-curve/

See Also

`cal_confus()`

Examples

```
data(test_data)
true_vec <- test_data[, 1]
pred_vec <- test_data[, 5]
confus_res <- cal_confus(true_vec, pred_vec)
AUC_res <- cal_auc(confus_res$TPR, confus_res$FPR)
```

`cal_confus` *Calculate confusion matrices*

Description

This function calculates the confusion matrices across different cutoff points.

Usage

```
cal_confus(true_vec, pred_vec, force_diag=TRUE)
```

Arguments

`true_vec`	A binary vector of real labels
`pred_vec`	A continuous predicted score(probabilities) vector, must be the same length with `true_vec`
`force_diag`	If TRUE, TPR and FPR will be forced to across (0, 0) and (1, 1)

Details

This function calculates the TP, FP, FN, TN, TPR, FPR and PPV across different cutoff points of `pred_vec`. TPR and FPR are forced to across (0, 0) and (1, 1) if `force_diag=TRUE`.

Value

`TP`	True positive
`FP`	False positive
`FN`	False negative
`TN`	True negative
`TPR`	True positive rate
`FPR`	False positive rate
`PPV`	Positive predictive value

References

https://en.wikipedia.org/wiki/Confusion_matrix

Examples

```
data(test_data)
true_vec <- test_data[, 1]
pred_vec <- test_data[, 5]
confus_res <- cal_confus(true_vec, pred_vec)
```

multi_pr *Multi-class classification PR*

Description

This function calculates the Precision, Recall and AUC of multi-class classifications.

Usage

```
multi_pr(data, force_diag=TRUE)
```

Arguments

data	A data frame contain true labels of multiple groups and corresponding predictive scores
force_diag	If TRUE, TPR and FPR will be forced to across (0, 0) and (1, 1)

Details

A data frame is required for this function as input. This data frame should contains true label (0 - Negative, 1 - Positive) columns named as XX_true (e.g. S1_true, S2_true and S3_true) and predictive scores (continuous) columns named as XX_pred_YY (e.g. S1_pred_SVM, S2_pred_RF), thus this function allows calcluating ROC on mulitiple classifiers.

Predictive scores could be probabilities among [0, 1] and other continuous values. For each classifier, the number of columns should be equal to the number of groups of true labels. The order of columns won't affect results.

Recall, Precision, AUC for each group and each method will be calculated. Macro/Micro-average AUC for all groups and each method will be calculated.

Micro-average ROC/AUC was calculated by stacking all groups together, thus converting the multi-class classification into binary classification. Macro-average ROC/AUC was calculated by averaging all groups results (one vs rest) and linear interpolation was used between points of ROC.

AUC will be calculated using function `cal_auc()`.

Value

Recall	A list of recalls for each group, each method and micro-/macro- average
Precision	A list of precisions for each group, each method and micro-/macro- average
AUC	A list of AUCs for each group, each method and micro-/macro- average
Methods	A vector contains the name of different classifiers
Groups	A vector contains the name of different groups

Examples

```
data(test_data)
pr_test <- multi_pr(test_data)
pr_test$AUC
```

`multi_roc` *Multi-class classification ROC*

Description

This function calculates the Specificity, Sensitivity and AUC of multi-class classifications.

Usage

```
multi_roc(data, force_diag=TRUE)
```

Arguments

`data`	A data frame contain true labels of multiple groups and corresponding predictive scores
`force_diag`	If TRUE, TPR and FPR will be forced to across (0, 0) and (1, 1)

Details

A data frame is required for this function as input. This data frame should contains true label (0 - Negative, 1 - Positive) columns named as XX_true (e.g. S1_true, S2_true and S3_true) and predictive scores (continuous) columns named as XX_pred_YY (e.g. S1_pred_SVM, S2_pred_RF), thus this function allows calcluating ROC on mulitiple classifiers.

Predictive scores could be probabilities among [0, 1] and other continuous values. For each classifier, the number of columns should be equal to the number of groups of true labels. The order of columns won't affect results.

Specificity, Sensitivity, AUC for each group and each method will be calculated. Macro/Micro-average AUC for all groups and each method will be calculated.

Micro-average ROC/AUC was calculated by stacking all groups together, thus converting the multi-class classification into binary classification. Macro-average ROC/AUC was calculated by averaging all groups results (one vs rest) and linear interpolation was used between points of ROC.

AUC will be calculated using function `cal_auc()`.

Value

`Specificity`	A list of specificities for each group, each method and micro-/macro- average
`Sensitivity`	A list of sensitivities for each group, each method and micro-/macro- average
`AUC`	A list of AUCs for each group, each method and micro-/macro- average
`Methods`	A vector contains the name of different classifiers
`Groups`	A vector contains the name of different groups

References

http://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html

Examples

```
data(test_data)
roc_test <- multi_roc(test_data)
roc_test$AUC
```

`plot_pr_data` *Generate PR plotting data*

Description

This function generates plotting PR data for following data visualization.

Usage

```
plot_pr_data(pr_res)
```

Arguments

`pr_res`	A list of results from multi_pr function.

Value

`pr_res_df`	The dataframe of results from multi_pr function, which is easy be visualized by ggplot2.

Examples

```
data(test_data)
pr_res <- multi_pr(test_data)
pr_res_df <- plot_pr_data(pr_res)
```

`plot_roc_data` *Generate ROC plotting data*

Description

This function generates plotting ROC data for following data visualization.

Usage

```
plot_roc_data(roc_res)
```

Arguments

`roc_res`	A list of results from multi_roc function.

Value

roc_res_df	The dataframe of results from multi_roc function, which is easy be visualized by ggplot2.

Examples

```
data(test_data)
roc_res <- multi_roc(test_data)
roc_res_df <- plot_roc_data(roc_res)
```

---

pr_auc_with_ci *Output of PR bootstrap confidence intervals*

Description

This function uses bootstrap to generate five types of equi-tailed two-sided confidence intervals of PR-AUC with different required percentages and output a dataframe with AUCs, lower CIs, and higher CIs of all methods and groups.

Usage

```
pr_auc_with_ci(data, conf= 0.95, type='bca', R = 100)
```

Arguments

data	A data frame contains true labels of multiple groups and corresponding predictive scores.
conf	A scalar contains the required level of confidence intervals, and the default number is 0.95.
type	A vector of character strings includes five different types of equi-tailed two-sided nonparametric confidence intervals (e.g., "norm","basic", "stud", "perc", "bca").
R	A scalar contains the number of bootstrap replicates, and the default number is 100.

Details

A data frame is required for this function as input. This data frame should contains true label (0 - Negative, 1 - Positive) columns named as XX_true (e.g. S1_true, S2_true and S3_true) and predictive scores (continuous) columns named as XX_pred_YY (e.g. S1_pred_SVM, S2_pred_RF). Predictive scores could be probabilities among [0, 1] and other continuous values. For each classifier, the number of columns should be equal to the number of groups of true labels. The order of columns won't affect results.

Value

norm	Using the normal approximation to calculate the confidence intervals.
basic	Using the basic bootstrap method to calculate the confidence intervals.
stud	Using the studentized bootstrap method to calculate the confidence intervals.
perc	Using the bootstrap percentile method to calculate the confidence intervals.
bca	Using the adjusted bootstrap percentile method to calculate the confidence intervals.

Examples

```
## Not run: data(test_data)
pr_auc_with_ci_res <- pr_auc_with_ci(test_data, conf= 0.95, type='bca', R = 100)
## End(Not run)
```

---

pr_ci *PR bootstrap confidence intervals*

Description

This function uses bootstrap to generate five types of equi-tailed two-sided confidence intervals of PR-AUC with different required percentages.

Usage

```
pr_ci(data, conf= 0.95, type='basic', R = 100, index = 4)
```

Arguments

data	A data frame contains true labels of multiple groups and corresponding predictive scores.
conf	A scalar contains the required level of confidence intervals, and the default number is 0.95.
type	A vector of character strings includes five different types of equi-tailed two-sided nonparametric confidence intervals (e.g., "norm","basic", "stud", "perc", "bca", "all").
R	A scalar contains the number of bootstrap replicates, and the default number is 100.
index	A scalar contains the position of the variable of interest.

Details

A data frame is required for this function as input. This data frame should contains true label (0 - Negative, 1 - Positive) columns named as XX_true (e.g. S1_true, S2_true and S3_true) and predictive scores (continuous) columns named as XX_pred_YY (e.g. S1_pred_SVM, S2_pred_RF). Predictive scores could be probabilities among [0, 1] and other continuous values. For each classifier, the number of columns should be equal to the number of groups of true labels. The order of columns won't affect results.

Value

norm	Using the normal approximation to calculate the confidence intervals.
basic	Using the basic bootstrap method to calculate the confidence intervals.
stud	Using the studentized bootstrap method to calculate the confidence intervals.
perc	Using the bootstrap percentile method to calculate the confidence intervals.
bca	Using the adjusted bootstrap percentile method to calculate the confidence intervals.
all	Using all previous bootstrap methods to calculate the confidence intervals.

Examples

```
## Not run: data(test_data)
pr_ci_res <- pr_ci(test_data, conf= 0.95, type='basic', R = 1000, index = 4)
## End(Not run)
```

---

roc_auc_with_ci *Output of ROC bootstrap confidence intervals*

Description

This function uses bootstrap to generate five types of equi-tailed two-sided confidence intervals of ROC-AUC with different required percentages and output a dataframe with AUCs, lower CIs, and higher CIs of all methods and groups.

Usage

```
roc_auc_with_ci(data, conf= 0.95, type='bca', R = 100)
```

Arguments

data	A data frame contains true labels of multiple groups and corresponding predictive scores.
conf	A scalar contains the required level of confidence intervals, and the default number is 0.95.
type	A vector of character strings includes five different types of equi-tailed two-sided nonparametric confidence intervals (e.g., "norm","basic", "stud", "perc", "bca").
R	A scalar contains the number of bootstrap replicates, and the default number is 100.

Details

A data frame is required for this function as input. This data frame should contains true label (0 - Negative, 1 - Positive) columns named as XX_true (e.g. S1_true, S2_true and S3_true) and predictive scores (continuous) columns named as XX_pred_YY (e.g. S1_pred_SVM, S2_pred_RF). Predictive scores could be probabilities among [0, 1] and other continuous values. For each classifier, the number of columns should be equal to the number of groups of true labels. The order of columns won't affect results.

Value

`norm`	Using the normal approximation to calculate the confidence intervals.
`basic`	Using the basic bootstrap method to calculate the confidence intervals.
`stud`	Using the studentized bootstrap method to calculate the confidence intervals.
`perc`	Using the bootstrap percentile method to calculate the confidence intervals.
`bca`	Using the adjusted bootstrap percentile method to calculate the confidence intervals.

Examples

```
## Not run: data(test_data)
roc_auc_with_ci_res <- roc_auc_with_ci(test_data, conf= 0.95, type='bca', R = 100)
## End(Not run)
```

`roc_ci` *ROC bootstrap confidence intervals*

Description

This function uses bootstrap to generate five types of equi-tailed two-sided confidence intervals of ROC-AUC with different required percentages.

Usage

```
roc_ci(data, conf= 0.95, type='basic', R = 100, index = 4)
```

Arguments

`data`	A data frame contains true labels of multiple groups and corresponding predictive scores.
`conf`	A scalar contains the required level of confidence intervals, and the default number is 0.95.
`type`	A vector of character strings includes five different types of equi-tailed two-sided nonparametric confidence intervals (e.g., "norm","basic", "stud", "perc", "bca", "all").
`R`	A scalar contains the number of bootstrap replicates, and the default number is 100.
`index`	A scalar contains the position of the variable of interest.

Details

A data frame is required for this function as input. This data frame should contains true label (0 - Negative, 1 - Positive) columns named as XX_true (e.g. S1_true, S2_true and S3_true) and predictive scores (continuous) columns named as XX_pred_YY (e.g. S1_pred_SVM, S2_pred_RF). Predictive scores could be probabilities among [0, 1] and other continuous values. For each classifier, the number of columns should be equal to the number of groups of true labels. The order of columns won't affect results.

Value

`norm`	Using the normal approximation to calculate the confidence intervals.
`basic`	Using the basic bootstrap method to calculate the confidence intervals.
`stud`	Using the studentized bootstrap method to calculate the confidence intervals.
`perc`	Using the bootstrap percentile method to calculate the confidence intervals.
`bca`	Using the adjusted bootstrap percentile method to calculate the confidence intervals.
`all`	Using all previous bootstrap methods to calculate the confidence intervals.

Examples

```
## Not run: data(test_data)
roc_ci_res <- roc_ci(test_data, conf= 0.95, type='basic', R = 1000, index = 4)
## End(Not run)
```

`test_data` *Example dataset*

Description

This example dataset contains two classifiers (m1, m2), and three groups (G1, G2, G3).

Usage

```
data("test_data")
```

Format

A data frame with 85 observations on the following 9 variables.

`G1_true` true labels of G1 (0 - Negative, 1 - Positive)

`G2_true` true labels of G2 (0 - Negative, 1 - Positive)

`G3_true` true labels of G3 (0 - Negative, 1 - Positive)

`G1_pred_m1` predictive scores of G1 in the classifier m1

`G2_pred_m1` predictive scores of G2 in the classifier m1

`G3_pred_m1` predictive scores of G3 in the classifier m1

`G1_pred_m2` predictive scores of G1 in the classifier m2

`G2_pred_m2` predictive scores of G2 in the classifier m2

`G3_pred_m2` predictive scores of G3 in the classifier m2

Examples

```
data(test_data)
```

Index