Package ‘PresenceAbsence’

March 2, 2022

Type Package
Title Presence-Absence Model Evaluation
Version 1.1.10
Date 2022-02-28
Depends R (>= 2.2.0)
Author Elizabeth Freeman
Maintainer Elizabeth Freeman <eafreeman@fs.fed.us>
Description Provides a set of functions useful when evaluating the results of presence-absence models. Package includes functions for calculating threshold dependent measures such as confusion matrices, pcc, sensitivity, specificity, and Kappa, and produces plots of each measure as the threshold is varied. It will calculate optimal threshold choice according to a choice of optimization criteria. It also includes functions to plot the threshold independent ROC curves along with the associated AUC (area under the curve).
License Unlimited
NeedsCompilation no
Repository CRAN
Date/Publication 2022-03-02 10:40:02 UTC

R topics documented:

PresenceAbsence-package ........................................ 2
auc ................................................................. 4
auc.roc.plot ...................................................... 5
calibration.plot .................................................. 9
cmx ................................................................. 11
error.threshold.plot ............................................ 13
Kappa ............................................................... 17
optimal.thresholds .............................................. 18
pcc ................................................................. 22
predicted.prevalence .......................................... 23
presence.absence.accuracy .................................... 24
presence.absence.hist ........................................ 26
PresenceAbsence-package

presence.absence.simulation ........................................... 29
presence.absence.summary ............................................. 31
roc.plot.calculate ....................................................... 35
sensitivity ............................................................... 37
SIM3DATA ................................................................. 38
SPDATA ................................................................. 38
specificity ............................................................ 39
SPPREV ............................................................... 40

Index 42

PresenceAbsence-package

Presence-Absence model evaluation

Description

Provides a set of functions useful when evaluating the results of presence-absence models. Package
includes functions for calculating threshold dependent measures such as confusion matrices, pcc,
sensitivity, specificity, and Kappa, and produces plots of each measure as the threshold is varied.
It also includes functions to plot the threshold independent ROC curves along with the associated
AUC (area under the curve).

Details

Package: PresenceAbsence
Type: Package
Version: 1.1.10
Date: 2022-02-28
License: This code was written and prepared by a U.S. Government employee on official time, and therefore it is in the publ...
sensitivity, specificity, Kappa, presence.absence.accuracy with find.auc set to false, predicted.prevalence, and the graphical function presence.absence.hist with N.bar set to 2.

Most functions take the dataframe of observed and predicted values (DATA) as input. The exceptions are the sub-functions that calculate single accuracy statistics: pcc, sensitivity, specificity, and Kappa. These sub-functions take the confusion matrix from cmx as input.

Some functions only evaluate one set of model predictions at a time, while others will work with multiple sets of model predictions. Even if the function only works on single models, the dataframe DATA can still contain multiple model predictions. Just use the argument which.model to indicate the desired column.

Functions that will only work on single models: cmx, auc, roc.plot.calculate, presence.absence.hist, error.threshold.plot, calibration.plot, and presence.absence.summary.

Functions that will work with multiple models: presence.absence.accuracy, optimal.thresholds, predicted.prevalence, and auc.roc.plot.

Note that this library provides graphical and tabular comparisons between models. It does not provide significance testing of model differences. The standard deviations given by presence.absence.accuracy are for each model individually. To test AUC for differences between models it is necessary to account for correlation. If you are interested in AUC significance testing, both pair-wise and overall, the Splus ROC library from Mayo clinic provides such a test. See auc for more details.

This code was written and prepared by a U.S. Government employee on official time, and therefore it is in the public domain and not subject to copyright.

Author(s)

Author: Elizabeth Freeman <elizabeth.a.freeman@usda.gov>
Maintainer: Elizabeth Freeman <eafreeman@fs.fed.us>

References


Examples

data(SIM3DATA)
auc.roc.plot(SIM3DATA)
presence.absence.summary(SIM3DATA, which.model=1)
Area Under the Curve

Description

auc calculates the area under the ROC curve approximated with a Mann-Whitney U statistic, and (optionally) the associated standard deviation.

Usage

\[
\text{auc(DATA, st.dev = TRUE, which.model = 1, na.rm = FALSE)}
\]

Arguments

- **DATA**: a matrix or dataframe of observed and predicted values where each row represents one plot and where columns are:
  - DATA[,1]: plot ID text
  - DATA[,2]: observed values zero-one values
  - DATA[,3]: predicted probabilities from first model numeric (between 0 and 1)
  - DATA[,4]: predicted probabilities from second model, etc...
- **st.dev**: a logical indicating whether the associated standard deviation should be calculated
- **which.model**: a number indicating which model from DATA should be used
- **na.rm**: a logical indicating whether missing values should be removed

Details

auc approximates the area under the ROC curve with a Mann-Whitney U statistic (Delong et al., 1988) to calculate the area under the curve.

The standard errors from auc are only valid for comparing an individual model to random assignment (i.e. AUC=.5). To compare two models to each other it is necessary to account for correlation due to the fact that they use the same test set. If you are interested in pair wise model comparisons see the Splus ROC library from Mayo clinic. auc is a much simpler function than what is available from the Splus ROC library from Mayo clinic.

The observed values (column 2 in DATA) can be given as 0/1 values to represent absence and presence. If this column contains actual values (i.e. basal area, biomass, etc...), any value of zero will be treated as absence and any value greater than zero will be treated as presence.

If observed values are all the same, in other words, if the data consists entirely of observed Presences or entirely of observed Absences, auc will return NaN.

Value

if st.dev = FALSE, returns: AUC area under the curve.
if st.dev = TRUE, returns a dataframe where:

[1,1]   AUC   area under the curve
[1,2]   AUC.sd  standard deviation of AUC

Author(s)

Elizabeth Freeman <eafreeman@fs.fed.us>

References


Splus ROC library developed by Beth Atkinson and Doug Mahoney at the Mayo Clinic is available at: http://www.stats.ox.ac.uk/pub/MASS3/Winlibs/ for windows.

See Also

cmx, pcc, sensitivity, specificity, Kappa, auc.roc.plot

Examples

data(SIM3DATA)

auc(SIM3DATA)

auc(SIM3DATA,st.dev=FALSE,which.model=2)

Description

auc.roc.plot creates a ROC plot for one dataset and one or more model predictions. Prints AUC for each model as part of the legend. auc.roc.plot also includes an option to mark several types of optimal thresholds along each ROC plot.

Usage

auc.roc.plot(DATA, threshold = 101, find.auc = TRUE, which.model = (1:(ncol(DATA) - 2)), na.rm = FALSE, xlab = "1-Specificity (false positives)", ylab = "Sensitivity (true positives)", main = "ROC Plot", model.names = NULL, color = NULL, line.type = NULL, lwd = 1, mark = 0, mark.numbers = TRUE, mark.color = NULL, opt.thresholds = NULL, opt.methods = NULL, req.sens, req.spec, obs.prev = NULL, smoothing = 1, add.legend = TRUE,
legend.text = model.names, legend.cex = 0.8, add.opt.legend = TRUE,
opt.legend.text = NULL, opt.legend.cex = 0.7,
counter.diagonal = FALSE, pch = NULL, FPC, FNC, cost.line = FALSE)

Arguments

DATA a matrix or dataframe of observed and predicted values where each row represents one plot and where columns are:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA[,1]</td>
<td>plot ID</td>
</tr>
<tr>
<td>DATA[,2]</td>
<td>observed values</td>
</tr>
<tr>
<td>DATA[,3]</td>
<td>predicted probabilities from first model</td>
</tr>
<tr>
<td>DATA[,4]</td>
<td>predicted probabilities from second model, etc...</td>
</tr>
</tbody>
</table>

threshold cutoff values between zero and one used for translating predicted probabilities into 0/1 values, defaults to 0.5. It can be a single value between zero and one, a vector of values between zero and one, or a positive integer representing the number of evenly spaced thresholds to calculate.

find.auc a logical indicating if area under the curve should be calculated

which.model a number indicating which model from DATA should be used

na.rm a logical indicating whether missing values should be removed

xlab a title for the x axis

ylab a title for the y axis

main an overall title for the plot

model.names a vector of the names of each model included in DATA to be used in the legend box

color should each model be plotted in a different color. It can be a logical value (where TRUE = color and FALSE = black and white), or a vector of color codes specifying particular colors for each line.

line.type should each model be plotted in a different line type. It can be a logical value (where TRUE = dashed lines and FALSE = solid lines), or a vector of codes specifying particular line types for each line.

lwd line width

mark particular thresholds to mark along each roc plot, given in same format as threshold. Note: if optimal.thresholds = TRUE, argument mark will be ignored.

mark.numbers a logical indication if the threshold values of each marked point along the ROC curved should be labeled next to the points

mark.color should the marked thresholds be plotted in a different color for each model. A logical value where TRUE equals same colors as the lines, and FALSE = marks are always black. Can also be specified as a vector of color codes. Note that in this case, it is one color per model, not one color per threshold.

opt.thresholds logical indicating whether the optimal thresholds should be calculated and plotted, or a vector specifying thresholds to plot

opt.methods what methods should be used to optimize thresholds. Argument can be given either as a vector of method names or method numbers. Possible values are:
1 Default threshold=0.5
2 Sens=Spec sensitivity=specificity
3 MaxSens+Spec maximizes (sensitivity+specificity)/2
4 MaxKappa maximizes Kappa
5 MaxPCC maximizes PCC (percent correctly classified)
6 PredPrev=Obs predicted prevalence=observed prevalence
7 ObsPrev threshold=observed prevalence
8 MeanProb mean predicted probability
9 MinROCdist minimizes distance between ROC plot and (0,1)
10 ReqSens user defined required sensitivity
11 ReqSpec user defined required specificity

req.sens a value between zero and one giving the user defined required sensitivity. Only used if opt.thresholds = TRUE. Note that req.sens = (1-maximum allowable errors for points with positive observations).

req.spec a value between zero and one giving the user defined required specificity. Only used if opt.thresholds = TRUE. Note that req.spec = (1- maximum allowable errors for points with negative observations).

obs.prev observed prevalence for opt.method = "PredPrev=Obs" and "ObsPrev". Defaults to observed prevalence from DATA.

smoothing smoothing factor for maximizing/minimizing. Only used if opt.thresholds = TRUE. Instead of find the threshold that gives the max/min value, function will average the thresholds of the given number of max/min values.

add.legend a logical indicating if a legend for AUC lines should be added to plot

legend.text a two item vector of text for presence/absence legend. Defaults to 'model.names'.

legend.cex cex for AUC legend

add.opt.legend logical indicating if a legend for optimal threshold criteria should be included on the plot

opt.legend.text a vector of text for optimalimal threshold criteria legend. Defaults to text corresponding to 'opt.methods'.

opt.legend.cex cex for optimization criteria legend

counter.diagonal should a counter-diagonal line be plotted. Note: each ROC plot crosses this line at the point where sensitivity equals specificity for that model.

pch plotting "character", i.e., symbol to use for the thresholds specified in mark. pch can either be a single character or an integer code for one of a set of graphics symbols. See help(points) for details.

FPC False Positive Costs, or for C/B ratio C = 'net costs of treating nondiseaseed individuals'.

FNC False Negative Costs, or for C/B ratio B = 'net benefits of treating diseased individuals'.

cost.line a logical indicating if the line representing the realtive cost ratio should be added to the plot.
Details

Receiver Operating Curves (ROC plots) provide a threshold independent method of evaluating the performance of presence/absence models. In a ROC plot the true positive rate (sensitivity) is plotted against the false positive rate (1.0-specificity) as the threshold varies from 0 to 1. A good model will achieve a high true positive rate while the false positive rate is still relatively small; thus the ROC plot will rise steeply at the origin, and then level off at a value near the maximum of 1. The ROC plot for a poor model (whose predictive ability is the equivalent of random assignment) will lie near the diagonal, where the true positive rate equals the false positive rate for all thresholds. Thus the area under the ROC curve (AUC) is a good measure of overall model performance, with good models having an AUC near 1, while poor models have an AUC near 0.5.

mark can be used to mark particular thresholds along each ROC plot, alternatively, if optimal.thresholds = TRUE the function will find optimal thresholds by several criteria and plot them along each ROC curve.

See optimal.thresholds for more details on the optimization methods, and on the arguments ReqSens, ReqSpec, obs.prev smoothing, FPC, FNC, and cost.line.

Note: if too many methods are included in opt.methods, the graph will get very crowded.

Value

creates a graphical plot

Author(s)

Elizabeth Freeman <eafreeman@fs.fed.us>

See Also

optimal.thresholds, presence.absence.accuracy, roc.plot.calculate, error.threshold.plot, presence.absence.summary

Examples

data(SIM3DATA)

auc.roc.plot(SIM3DATA)

auc.roc.plot(SIM3DATA, opt.thresholds=TRUE, opt.methods=c("Default","Sens=Spec","MinROCdist"))

auc.roc.plot(SIM3DATA, threshold=101, which.model=c(2,3), model.names=c("model a","model b","model c"), na.rm=TRUE, xlab="1-Specificity (false positives)", ylab="Sensitivity (true positives)", main="ROC Plot", color=TRUE)
`calibration.plot` produces a goodness-of-fit plot for Presence/Absence data.

**Usage**

```r
calibration.plot(DATA, which.model = 1, na.rm = FALSE, alpha = 0.05, N.bins = 5, 
xlab = "Predicted Probability of Occurrence",  
ylab = "Observed Occurrence as Proportion of Sites Surveyed",  
main = NULL, color = NULL, model.names = NULL)
```

**Arguments**

- **DATA** a matrix or dataframe of observed and predicted values where each row represents one plot and where columns are:
  - `DATA[,1]` plot ID text
  - `DATA[,2]` observed values zero-one values
  - `DATA[,3]` predicted probabilities from first model numeric (between 0 and 1)
  - `DATA[,4]` predicted probabilities from second model, etc...

- **which.model** a number indicating which model from DATA should be used

- **na.rm** a logical indicating whether missing values should be removed

- **alpha** alpha value for confidence intervals

- **N.bins** number of bins to split predicted probabilities into

- **xlab** a title for the x axis
calibration.plot

ylab               a title for the y axis
main               an overall title for the plot
color              a logical or a vector of color codes
model.names        a vector of the names of each model included in DATA

Details

Takes a single model and creates a goodness-of-fit plot of observed verses predicted values. The plots are grouped into bins based on their predicted values, and then the bin prevalence (the ratio of plots in this bin with observed values of present verses the total number of plots in this bin) is calculated for each bin. The confidence interval for each bin is also plotted, and the total number of plots is labeled above each the bin.

Confidence intervals are calculated for the binomial bin counts using the F distribution.

Unlike a typical goodness-of-fit plot from a linear regression model, with Presence/Absence data having all the points lay along the diagonal does not necessarily imply a good quality model. The ideal calibration plot for Presence/Absence data depends on the intended use of the model.

If the model is to be used to produce probability maps, then it is indeed desirable that (for example) 80 percent of plots with predicted probability of 0.8 actually do have observed Presence. In this case, having all the bins along the diagonal does indicate a good model.

However, if model is to be used simply to predict species presence, then all that is required is that some threshold exists (not necessarily 0.5) where every plot with a lower predicted probability is observed Absent, and every plot with a higher predicted probability is observed Present. In this case, a good model will not necessarily (in fact, will rarely) have all the bins along the diagonal. (Note: for this purpose presence.absence.hist may produce more useful diagnostics.)

If all the bins lie above the diagonal, or all the bins lie below the diagonal, it may indicate that the training and test datasets have different prevalence. In this case, it may be worthwhile to re-examine the initial data selection.

Value

creates a graphical plot
returns a dataframe of information about the bins where:

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>BinCenter</td>
</tr>
<tr>
<td>[2]</td>
<td>NBin</td>
</tr>
<tr>
<td>[3]</td>
<td>BinObs</td>
</tr>
<tr>
<td>[4]</td>
<td>BinPred</td>
</tr>
<tr>
<td>[5]</td>
<td>BinObsCIlower</td>
</tr>
<tr>
<td>[6]</td>
<td>BinObsCIupper</td>
</tr>
</tbody>
</table>

Author(s)

Elizabeth Freeman <eafreeman@fs.fed.us>
References


See Also

presence.absence.summary, presence.absence.hist

Examples

data(SIM3DATA)

calibration.plot(SIM3DATA)

calibration.plot( DATA=SIM3DATA,
which.model=3,
na.rm=TRUE,
alpha=0.05,
N.bins=10,
 xlab="Predicted Probability of Occurrence",
 ylab="Observed occurrence as proportion of sites surveyed",
model.names=NULL,
main=NULL)

---

cmx

Confusion Matrix

Description

cmx calculates the confusion matrix for a single model.

Usage

`cmx(DATA, threshold = 0.5, which.model = 1, na.rm = FALSE)`

Arguments

DATA 
a matrix or dataframe of observed and predicted values where each row represents one plot and where columns are:

<table>
<thead>
<tr>
<th>DATA[,1]</th>
<th>plot ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA[,2]</td>
<td>observed values</td>
</tr>
<tr>
<td>DATA[,3]</td>
<td>predicted probabilities from first model</td>
</tr>
<tr>
<td>DATA[,4]</td>
<td>predicted probabilities from second model, etc...</td>
</tr>
</tbody>
</table>

threshold 
a cutoff value between zero and one used for translating predicted probabilities
into 0/1 values, defaults to 0.5. It must be a single value between zero and one.

which.model  a number indicating which model from DATA should be used
na.rm        a logical indicating whether missing values should be removed

Details

cmx calculates the confusion matrix for a single model at a single threshold.
If DATA contains more predictions from more than one model WHICH.DATA can be used to specify which model should be used. If WHICH.DATA is not given, cmx will use predictions from the first model by default.

When calculating the confusion matrix, any plot with a predicted probability greater than threshold is considered to be predicted Present, while any plot with a predicted probability less than or equal to threshold is considered to be predicted Absent. The only exception is when threshold equals zero. In that case, all plots are considered to be predicted Present.

Unlike other functions in this library, threshold cannot be a vector or an integer greater than one. Instead, threshold must be given as a single number between zero and one.
If na.rm equals FALSE and NA's are present in the DATA function will return NA.
If na.rm equals TRUE and NA's are present in the DATA, function will remove all rows where any of the values in the row consist of NA. Function will also print the number of rows that have been removed.

Value

the confusion matrix is returned in the form of a table where:

columns     observed values
rows        predicted values

Author(s)

Elizabeth Freeman <eafreeman@fs.fed.us>

See Also

pcc, sensitivity, specificity, Kappa

Examples

### EXAMPLE 1 ###
## generate simulated data ##
set.seed(666)
N=1000
SIMDATA<-matrix(0,N,3)
SIMDATA<-as.data.frame(SIMDATA)
names(SIMDATA)<-c("plotID","Observed","Predicted")
SIMDATA$plotID<-1:N
SIMDATA$Observed<-rbinom(n=N,size=1,prob=.2)
error.threshold.plot

SIMDATA$Predicted[SIMDATA$Observed==1]<-rnorm(n=length(SIMDATA$Observed[SIMDATA$Observed==1]), mean=8, sd=.15)
SIMDATA$Predicted[SIMDATA$Observed==0]<-rnorm(n=length(SIMDATA$Observed[SIMDATA$Observed==0]), mean=2, sd=.15)
SIMDATA$Predicted<-((SIMDATA$Predicted-min(SIMDATA$Predicted))/(max(SIMDATA$Predicted)-min(SIMDATA$Predicted)))

### plot simulated data
hist(SIMDATA$Predicted,100)

### calculate confusion matrix ###
cmx(SIMDATA)

### EXAMPLE 2 ###
data(SIM3DATA)
cmx(SIM3DATA)
cmx(SIM3DATA,which.model=2)
cmx(SIM3DATA,which.model=3,threshold=.2)

---

error.threshold.plot  Error Threshold Plot

Description

error.threshold.plot takes a single model and plots the sensitivity and specificity as a function of threshold. It will optionally add other error statistics such as PCC and/or Kappa to the plot. Optionally, it will also optimize the choice of threshold by several criteria, return the results as a dataframe, and mark the optimized thresholds on the plot.

Usage

error.threshold.plot(DATA, threshold = 101, which.model = 1, na.rm = FALSE, xlab = "Threshold", ylab = "Accuracy Measures", main = NULL, model.names = NULL, color = NULL, line.type = NULL, lwd = 1, plot.it = TRUE, opt.thresholds = NULL, opt.methods = NULL, req.sens, req.spec, obs.prev = NULL, smoothing = 1, vert.lines = FALSE, add.legend = TRUE, legend.text = legend.names, legend.cex = 0.8, add.opt.legend = TRUE, opt.legend.text = NULL, opt.legend.cex = 0.7, pch = NULL, FPC, FNC)

Arguments

DATA a matrix or dataframe of observed and predicted values where each row represents one plot and where columns are:

DATA[,1] plot ID text
DATA[,2] observed values zero-one values
DATA[,3] predicted probabilities from first model numeric (between 0 and 1)
DATA[,4] predicted probabilities from second model, etc...
threshold
cutoff values between zero and one used for translating predicted probabilities into 0 / 1 values, defaults to 0.5. It can be a single value between zero and one, a vector of values between zero and one, or a positive integer representing the number of evenly spaced thresholds to calculate.

which.model
a number indicating which model from DATA should be used

na.rm
a logical indicating whether missing values should be removed

xlab
a title for the x axis

ylab
a title for the y axis

main
an overall title for the plot

model.names
a vector of the names of each model included in DATA to be used in the legend box

color
should each error statistic be plotted in a different color. It can be a logical value (where TRUE = color and FALSE = black and white), or a vector of color codes specifying particular colors for each line.

line.type
should each model be plotted in a different line type. It can be a logical value (where TRUE = dashed lines and FALSE = solid lines), or a vector of codes specifying particular line types for each line.

lwd
line width

plot.it
a logical indicating if a graphical plot should be produced

opt.thresholds
logical indicating whether the optimal thresholds should be calculated and plotted, or a vector specifying thresholds to plot

opt.methods
what methods should be used to optimize thresholds. Given either as a vector of method names or method numbers. Possible values are:

1 Default threshold=0.5
2 Sens=Spec sensitivity=specificity
3 MaxSens+Spec maximizes (sensitivity+specificity)/2
4 MaxKappa maximizes Kappa
5 MaxPCC maximizes PCC (percent correctly classified)
6 PredPrev=Obs predicted prevalence=observed prevalence
7 ObsPrev threshold=observed prevalence
8 MeanProb mean predicted probability
9 MinROCdist minimizes distance between ROC plot and (0,1)
10 ReqSens user defined required sensitivity
11 ReqSpec user defined required specificity
12 Cost user defined relative costs ratio

req.sens
a value between zero and one giving the user defined required sensitivity. Only used if opt.thresholds = TRUE. Note that req.sens = (1-maximum allowable errors for points with positive observations).

req.spec
a value between zero and one giving the user defined required specificity. Only used if opt.thresholds = TRUE. Note that req.spec = (1-maximum allowable errors for points with negative observations).
error.threshold.plot

obs.prev observed prevalence for opt.method = "PredPrev=Obs" and "ObsPrev". Defaults to observed prevalence from DATA.

smoothing smoothing factor for maximizing/minimizing. Only used if opt.thresholds = TRUE. Instead of find the threshold that gives the max/min value, function will average the thresholds of the given number of max/min values.

vert.lines a logical where: TRUE means vertical lines added to plot at optimal thresholds; FALSE means no vertical lines, instead optimal thresholds marked along error statistics plots. Only used if opt.thresholds = TRUE.

add.legend logical indicating if a legend for accuracy statistics should be included on the plot

legend.text a vector of text for accuracy statistics legend. Defaults to name of each accuracy statistic.

legend.cex cex for presence/absence legend

add.opt.legend logical indicating if a legend for optimal threshold criteria should be included on the plot

opt.legend.text a vector of text for optimal threshold criteria legend. Defaults to text corresponding to 'opt.methods'.

opt.legend.cex cex for optimization criteria legend

pch plotting "character", i.e., symbol to use for the thresholds specified in MARK. pch can either be a single character or an integer code for one of a set of graphics symbols. See help(points) for details.

FPC False Positive Costs, or for C/B ratio C = 'net costs of treating nondiseased individuals'.

FNC False Negative Costs, or for C/B ratio B = 'net benefits of treating diseased individuals'.

Details

error.threshold.plot serves two purposes. First, if plot.it = TRUE, it produces a graphical plot. Second, if opt.thresholds = TRUE it will find optimal thresholds by several criteria. These optimal thresholds, along with basic accuracy measures for each type of optimal threshold will be returned as a dataframe. If a plot is produced, these optimal thresholds will be added to the plot.

The graphical plot will always include lines showing sensitivity and specificity as a function of threshold. In addition, for opt.methods = "MaxKappa", "MaxPCC", "MinRODdist", or "MaxSens+Spec" additional lines will be added to show the statistic being maximized/minimized.

These lines will be added to graph even if opt.thresholds = FALSE. So for example, to produce a graph showing sensitivity, specificity, and Kappa as functions of threshold, with out marking the optimal thresholds, set opt.thresholds = FALSE, and opt.methods = "MaxKappa".

See optimal.thresholds for more details on the optimization methods, and on the arguments ReqSens, ReqSpec, obs.prev, smoothing, FPC, and FNC.

When opt.thresholds = TRUE, the default is to plot the optimal thresholds directly along the corresponding error statistics (or along the sensitivity line if the method has no corresponding error statistic). If the argument vert.lines = TRUE, a vertical line is drawn at each optimal threshold, and the lines are labeled across the top of the plot.

Note: if too many methods are included in opt.methods, the graph will get very crowded.
error.threshold.plot

Value

If plot.it = TRUE creates a graphical plot.
If opt.thresholds = TRUE, returns a dataframe of information about the optimal thresholds where:

<table>
<thead>
<tr>
<th>[,1] legend.names</th>
<th>type of optimal threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>[,2] threshold</td>
<td>optimal threshold</td>
</tr>
<tr>
<td>[,3] PCC</td>
<td>at that threshold</td>
</tr>
<tr>
<td>[,4] sensitivity</td>
<td>at that threshold</td>
</tr>
<tr>
<td>[,5] specificity</td>
<td>at that threshold</td>
</tr>
<tr>
<td>[,6] Kappa</td>
<td>at that threshold</td>
</tr>
</tbody>
</table>

Author(s)

Elizabeth Freeman <eafreeman@fs.fed.us>

See Also

optimal.thresholds, presence.absence.accuracy, roc.plot.calculate, presence.absence.summary

Examples

data(SIM3DATA)

error.threshold.plot(SIM3DATA, opt.methods=c(1,2,5))

error.threshold.plot( SIM3DATA,
  which.model=2,
  opt.thresholds=TRUE,
  opt.methods=c("Default", "Sens=Spec", "MinROCdist"),
  vert.lines=TRUE)

error.threshold.plot( SIM3DATA,
  threshold=101,
  which.model=2,
  na.rm=TRUE,
  xlab="Threshold",
  ylab="Accuracy Measures",
  main="Error Rate verses Threshold",
  model.names=NULL,
  pch=NULL,
  color= c(3,5,7),
  line.type=NULL,
  lwd=1,
  plot.it=TRUE,
  opt.thresholds=TRUE,
  opt.methods=1:4,
  req.sens=0.85,
  req.spec=0.85,
Kappa

```r
obs.prev=NULL,
smoothing=1,
vert.lines=FALSE,
add.legend=TRUE,
legend.cex=0.8)
```

---

**Description**

Kappa calculates Kappa and (optionally) the associated standard deviation from a confusion matrix.

**Usage**

```r
Kappa(CMX, st.dev = TRUE)
```

**Arguments**

- `CMX`: a confusion matrix - output from `cmx`
- `st.dev`: a logical indicating whether the associated standard deviation should be calculated

**Details**

The Kappa statistic summarizes all the available information in the confusion matrix. Kappa measures the proportion of correctly classified units after accounting for the probability of chance agreement.

**Value**

If `st.dev = FALSE`, returns: Kappa.

If `st.dev = TRUE`, returns a dataframe where:

```r
[1,1] Kappa
[1,2] Kappa.sd standard deviation of Kappa
```

**Author(s)**

Elizabeth Freeman <eafreeman@fs.fed.us>

**See Also**

- `cmx`, `pcc`, `sensitivity`, `specificity`, `auc`
optimal.thresholds  

**Calculate Optimal Thresholds**

**Description**

optimal.thresholds calculates optimal thresholds for Presence/Absence data by any of several methods.

**Usage**

optimal.thresholds(DATA = NULL, threshold = 101, which.model = 1:(ncol(DATA)-2), model.names = NULL, na.rm = FALSE, opt.methods = NULL, req.sens, req.spec, obs.prev = NULL, smoothing = 1, FPC, FNC)

**Arguments**

- **DATA** a matrix or dataframe of observed and predicted values where each row represents one plot and where columns are:
  - `DATA[,1]` plot ID text
  - `DATA[,2]` observed values zero-one values
  - `DATA[,3]` predicted probabilities from first model numeric (between 0 and 1)
  - `DATA[,4]` predicted probabilities from second model, etc...

- **threshold** cutoff values between zero and one used for translating predicted probabilities into 0 / 1 values, defaults to 0.5. It can be a single value between zero and one, a vector of values between zero and one, or a positive integer representing the number of evenly spaced thresholds to calculate. To get reasonably good optimizations, there should be a large number of thresholds.

- **which.model** a number or vector indicating which models from `DATA` should be used

- **model.names** a vector of the names of each model included in `DATA` to be used as column names

- **na.rm** a logical indicating whether missing values should be removed

- **opt.methods** what methods should be used to optimize thresholds. Given either as a vector of method names or method numbers. Possible values are:
  - 1 Default threshold=0.5
  - 2 Sens=Spec sensitivity=specificity
  - 3 MaxSens+Spec maximizes (sensitivity+specificity)/2

**Examples**

```r
data(SIM3DATA)
Kappa(cmx(SIM3DATA))
Kappa(cmx(SIM3DATA),st.dev=FALSE)
```
optimal.thresholds

4 MaxKappa maximizes Kappa
5 MaxPCC maximizes PCC (percent correctly classified)
6 PredPrev=Obs predicted prevalence=observed prevalence
7 ObsPrev threshold=observed prevalence
8 MeanProb mean predicted probability
9 MinROCdist minimizes distance between ROC plot and (0,1)
10 ReqSens user defined required sensitivity
11 ReqSpec user defined required specificity
12 Cost user defined relative costs ratio

req.sens a value between zero and one giving the user defined required sensitivity. Only used if opt.thresholds = TRUE. Note that req.sens = (1-maximum allowable errors for points with positive observations).

req.spec a value between zero and one giving the user defined required specificity. Only used if opt.thresholds = TRUE. Note that req.sens = (1- maximum allowable errors for points with negative observations).

obs.prev observed prevalence for opt.method = "PredPrev=Obs" and "ObsPrev". Defaults to observed prevalence from DATA.

smoothing smoothing factor for maximizing/minimizing. Only used if opt.thresholds = TRUE. Instead of find the threshold that gives the max/min value, function will average the thresholds of the given number of max/min values.

FPC False Positive Costs, or for C/B ratio C = 'net costs of treating nondiseased individuals'.

FNC False Negative Costs, or for C/B ratio B = 'net benefits of treating diseased individuals'.

Details

The 'opt.methods' argument is allows the user to choose optimization methods. The methods can be specified by number (opt.methods = 1:12 or opt.methods = c(1,2,4)) or by name (opt.methods = c("Default","Sens=Spec","MaxKappa")).

There are currently twelve optimization criteria available:

"Default" First, the default criteria of setting 'threshold = 0.5'

"Sens=Spec" The second criteria for optimizing threshold choice is by finding the threshold where sensitivity equals specificity. In other words, find the threshold where positive observations are just as likely to be wrong as negative observations.

Note: when threshold is optimized by criteria "Sens=Spec" it is correlated to prevalence, so that rare species are given much lower thresholds than widespread species. As a result, rare species may give the appearance of inflated distribution, if maps are made with thresholds that have been optimized by this method (Manel, 2001).

"MaxSens+Spec" The third criteria chooses the threshold that maximizes the sum of sensitivity and specificity. In other words, it is minimizing the mean of the error rate for positive observations and the error rate for negative observations. This is equivalent to maximizing (sensitivity + specificity - 1), otherwise know as the Youden’s index, or the True Skill Statistic. Note that while Youden’s Index is independent of prevalence, using Youden’s index to select a threshold does have an effect on the predicted prevalence, causing the distribution of rare species to be over predicted.
"MaxKappa" The forth criteria for optimizing the threshold choice is to find the threshold that gives the maximum value of Kappa. Kappa makes full use of the information in the confusion matrix to assess the improvement over chance prediction.

"MaxPCC" The fifth criteria is to maximize the total accuracy (PCC - Percent Correctly Classified). Note: It may seem like maximizing total accuracy would be the obvious goal, however, there are many problems with using PCC to assess model accuracy. For example, with species with very low prevalence, it is possible to maximize PCC simply by declaring the species a absent at all locations – not a very useful prediction!

"PredPrev=Obs" The sixth criteria is to find the threshold where the Predicted prevalence is equal to the Observed prevalence. This is a useful method when preserving prevalence is of prime importance.

"ObsPrev" The seventh criteria is an even simpler variation, where you simply set the threshold to the Observed prevalence. It is nearly as good as method six at preserving prevalence and requires no computation.

"MeanProb" The eighth criteria also requires no threshold computation. Method eight sets the threshold to the mean probability of occurrence from the model results.

"MinROCdist" The ninth criteria is to find the threshold that minimizes the distance between the ROC plot and the upper left corner of the unit square.

"ReqSens" The tenth criteria allows the user to set a required sensitivity, and then finds the highest threshold that will meet this requirement. In other words, the user can decide that the model must miss no more than, for example 15 percent of the plots where the species is observed to be present. Therefore they require a sensitivity of at least 0.85. This may be useful if, for example, the goal is to define a management area for a rare species, and they want to be certain that the management area doesn’t leave unprotected too many populations.

"ReqSpec" The eleventh criteria allows the user to set a required specificity, and then finds the lowest threshold that will meet this requirement. In other words, the user can decide that the model must miss no more than, for example 15 percent of the plots where the species is observed to be absent. Therefore they require a specificity of at least 0.85. This may be useful if, for example, the goal is to determine if a species is threatened, and they want to be certain not to over inflate the population by over declaring true absences as predicted presences.

Note: for "ReqSens" and "ReqSpec", if your model is poor, and your requirement is too strict, it is possible that the only way to meet this requirement. In other words, the user can decide that the model must miss no more than, for example 15 percent of the plots where the species is observed to be absent. Therefore they require a specificity of at least 0.85. This may be useful if, for example, the goal is to determine if a species is threatened, and they want to be certain not to over inflate the population by over declaring true absences as predicted presences.

"Cost" The twelth criteria balances the relative costs of false positive predictions and false negative predictions. A slope is calculated as \( \frac{(FPC/FNC)((1 - prevalence)/prevalence)}{ } \). To determine the threshold, a line of this slope is moved from the top left of the ROC plot, till it first touches the ROC curve.

Note: the criteria "Cost" can also be used for C/B ratio analysis of diagnostic tests. In this case \( FPC = C \) (the net costs of treating nondiseased individuals) and \( FNC = B \) (the net benefits of treating diseased individuals). For further information on "Cost" see Wilson et. al. (2005) and Cantor et. al. (1999).
For all the criteria that depend on observed prevalence ("PredPrev=Obs", "ObsPrev" and cost), the default is to use the observed prevalence from DATA. However, the argument obs.prev can be used to substitute a predetermined value for observed prevalence, for example, the prevalence from a larger dataset.

error.threshold.plot is a rough and ready function. It optimizes thresholds simply by calculating a large number of evenly spaced thresholds and looking for the best ones. This is good enough for graphs, but to find the theoretically 'best' thresholds, would require calculating every possible unique threshold (not necessarily evenly spaced!).

Details on smoothing argument: when the statistic being maximized (e.g. Kappa) is relatively flat but erratic, just picking the threshold that gives single maximum value is somewhat arbitrary. smoothing compensates for this by taking an average of the thresholds that give a set number of the highest values (e.g. the 10 highest Kappa’s, or the 20 highest Kappa’s).

Value

If DATA is not provided function will return a vector of the possible optimization methods. Otherwise, returns a dataframe where:

<table>
<thead>
<tr>
<th></th>
<th>Method - names of optimization methods</th>
<th>optimal thresholds for the first model</th>
<th>optimal thresholds for the second model, etc...</th>
</tr>
</thead>
</table>

Author(s)

Elizabeth Freeman <eafreeman@fs.fed.us>

References


See Also

error.threshold.plot, presence.absence.accuracy, roc.plot.calculate, presence.absence.summary

Examples

data(SIM3DATA)
optimal.thresholds(SIM3DATA)
Description

pcc calculates the percent correctly classified and (optionally) the associated standard deviation from a confusion matrix.

Usage

pcc(CMX, st.dev = TRUE)

Arguments

CMX             a confusion matrix - output from cmx
st.dev          a logical indicating whether the associated standard deviation should be calculated

Details

Percent Correctly Classified is simply the proportion of test observations that are correctly classified.

Value

if st.dev = FALSE, returns: PCC percent correctly classified.

if st.dev = TRUE, returns a dataframe where:

<table>
<thead>
<tr>
<th>[1,1]</th>
<th>PCC</th>
<th>percent correctly classified</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,2]</td>
<td>PCC.sd</td>
<td>standard deviation of PCC</td>
</tr>
</tbody>
</table>

Author(s)

Elizabeth Freeman <eafreeman@fs.fed.us>

See Also

cmx, sensitivity, specificity, Kappa, auc

Examples

data(SIM3DATA)
pcc(cmx(SIM3DATA))
pcc(cmx(SIM3DATA), st.dev=FALSE)
**predicted.prevalence**  
*Predicted Prevalence*

**Description**

`predicted.prevalence` calculates the observed prevalence and predicted prevalence for one or more models at one or more thresholds.

**Usage**

`predicted.prevalence(DATA, threshold = 0.5, which.model = (1:N.models), na.rm = FALSE)`

**Arguments**

- **DATA**
  - a matrix or dataframe of observed and predicted values where each row represents one plot and where columns are:

    - DATA[,1] plot ID text
    - DATA[,2] observed values zero-one values
    - DATA[,3] predicted probabilities from first model numeric (between 0 and 1)
    - DATA[,4] predicted probabilities from second model, etc...

- **threshold**
  - a cutoff values between zero and one used for translating predicted probabilities into 0 / 1 values, defaults to 0.5. `threshold` can be a single value between zero and one, a vector of values between zero and one, or a positive integer representing the number of evenly spaced thresholds to calculate.

- **which.model**
  - a number indicating which models from DATA should be used

- **na.rm**
  - a logical indicating whether missing values should be removed

**Details**

Function will work for one model and multiple thresholds, or one threshold and multiple models, or multiple models each with their own threshold.

**Value**

returns a dataframe where:

- [,1] threshold thresholds used for each row in the table
- [,2] Obs.Prevalence this will be the same in each row
- [,3] Model 1 Predicted prevalence for first model
- [,4] Model 2 Predicted prevalence for second model, etc...

**Author(s)**

Elizabeth Freeman <eafreeman@fs.fed.us>
Examples

data(SIM3DATA)
predicted.prevalence(SIM3DATA)
predicted.prevalence(SIM3DATA, threshold=11, which.model=1, na.rm=FALSE)
predicted.prevalence(SIM3DATA, threshold=c(.2,.5,.7), na.rm=FALSE)

presence.absence.accuracy

Accuracy Table for Presence/Absence Data

Description

Calculates five accuracy measures (pcc, sensitivity, specificity, Kappa, and AUC) for Presence/Absence data, and (optionally) their associated standard deviations.

Usage

presence.absence.accuracy(DATA, threshold = 0.5, find.auc = TRUE, st.dev = TRUE, which.model = (1:(ncol(DATA) - 2)), na.rm = FALSE)

Arguments

DATA a matrix or dataframe of observed and predicted values where each row represents one plot and where columns are:

| DATA[,1] | plot ID | text |
| DATA[,2] | observed values | zero-one values |
| DATA[,3] | predicted probabilities from first model | numeric (between 0 and 1) |
| DATA[,4] | predicted probabilities from second model, etc... |

threshold a cutoff values between zero and one used for translating predicted probabilities into 0 /1 values, defaults to 0.5. If calculations are to be performed on a single model prediction threshold can be a single value between zero and one, a vector of values between zero and one, or a positive integer representing the number of evenly spaced thresholds to calculate. If calculations are to be performed on multiple model predictions, then threshold must be either a single value between zero and one, or a vector of the same length as the number of models.

find.auc a logical indicating if area under the curve should be calculated

st.dev a logical indicating if standard deviations should be calculated

which.model a number indicating which models from DATA should be used

na.rm a logical indicating whether missing values should be removed
presence.absence.accuracy

Details

presence.absence.accuracy calculates five standard accuracy measures for presence absence data, and (optionally) their associated standard deviations.

Function will work for one model and multiple thresholds, or one threshold and multiple models, or multiple models each with their own threshold.

Depending on the size of the dataset and the speed of the computer this function may take a couple of minutes to run. Finding the AUC is the slowest part of this function. The AUC can be suppressed by setting find.auc = FALSE.

which.model can be used to specify which of the prediction models from DATA should be used.

Value

if st.dev = FALSE, returns a dataframe where:

| .[1] | model | model name (column name from DATA) |
| .[2] | threshold | thresholds used for each row in the table |
| .[3] | PCC | percent correctly classified |
| .[4] | sensitivity | |
| .[5] | specificity | |
| .[6] | Kappa | |
| .[7] | AUC | area under the curve |

if st.dev = TRUE, returns a dataframe where:

| .[1] | model | model name (column name from DATA) |
| .[2] | threshold | thresholds used for each row in the table |
| .[3] | PCC | percent correctly classified |
| .[4] | sensitivity | |
| .[5] | specificity | |
| .[6] | Kappa | |
| .[7] | AUC | area under the curve |
| .[8] | PCC.sd | standard deviation of PCC |
| .[9] | sensitivity.sd | standard deviation of sensitivity |
| .[10] | specificity.sd | standard deviation of specificity |
| .[11] | Kappa.sd | standard deviation of Kappa |
| .[12] | AUC.sd | standard deviation of AUC |

if find.auc = FALSE, then columns for AUC and AUC.sd are not returned.

Author(s)

Elizabeth Freeman <eafreeman@fs.fed.us>

See Also

cmx, pcc, sensitivity, specificity, Kappa, auc
Examples

```r
data(SIM3DATA)

### EXAMPLE 1 - multiple model predictions at one threshold ###
presence.absence.accuracy(SIM3DATA)
presence.absence.accuracy(SIM3DATA, threshold=.4, st.dev=FALSE)
presence.absence.accuracy(SIM3DATA, which.model=c(1,3), st.dev=FALSE)

### EXAMPLE 2 - one model prediction at multiple thresholds ###
presence.absence.accuracy(SIM3DATA, threshold=c(.25,.5,.75), which.model=3)
presence.absence.accuracy(SIM3DATA, threshold=11, which.model=2)

### EXAMPLE 3 - multiple model predictions, each at its own threshold ###
presence.absence.accuracy(SIM3DATA, threshold=c(.5,.5,.2), which.model=c(1,2,2))
```

---

**presence.absence.hist**  
*Presence/Absence Histogram*

### Description ###

Produces a histogram of predicted probabilities with each bar subdivided by observed values.  `presence.absence.hist` also includes an option to mark several types of optimal thresholds along each plot.

### Usage ###

```r
presence.absence.hist(DATA, which.model = 1, na.rm = FALSE,
xlab = "predicted probability", ylab = "number of plots",
main = NULL, model.names = NULL, color = NULL, N.bars = 20,
truncate.tallest = FALSE, ylim = 1.25 * range(0, apply(counts, 2, sum)),
opt.thresholds = NULL, threshold = 101, opt.methods = NULL,
req.sens, req.spec, obs.prev = NULL, smoothing = 1, add.legend = TRUE,
legend.text=c("present","absent"), legend.cex = 0.8, add.opt.legend = TRUE,
opt.legend.text = NULL, opt.legend.cex = 0.7, pch = NULL, FPC, FNC)
```

### Arguments ###

- **DATA**
  - a matrix or dataframe of observed and predicted values where each row represents one plot and where columns are:
  - `DATA[,1]` plot ID
  - `DATA[,2]` observed values
  - `DATA[,3]` predicted probabilities from first model
  - `DATA[,4]` predicted probabilities from second model, etc...
presence.absence.hist

which.model  a number indicating which model from DATA should be used
na.rm        a logical indicating whether missing values should be removed
xlab         a title for the x axis
ylab         a title for the y axis
main         an overall title for the plot
model.names  a vector of the names of each model included in DATA
color        colors for presence/absence. Defaults to Presence = dark gray, Absence = light gray.
N.bars       number of bars in histogram
truncate.tallest a logical indicating if the tallest bar should be truncated to fit on plot
ylim         limit for y axis. To allow room for legend box ylim should be somewhat larger than largest bar.
opt.thresholds a logical indicating whether the optimal thresholds should be calculated and plotted, or a vector specifying thresholds to plot
threshold     cutoff values between zero and one used for translating predicted probabilities into 0/1 values, defaults to 0.5. It can be a single value between zero and one, a vector of values between zero and one, or a positive integer representing the number of evenly spaced thresholds to calculate. To get reasonably good optimizations, there should be a large number of thresholds. (Only used if opt.thresholds = TRUE.
opt.methods   what methods should be used to optimize thresholds. Argument can be given either as a vector of method names or method numbers. Possible values are:

1  Default    threshold=0.5
2  Sens=Spec  sensitivity=specificity
3  MaxSens+Spec maximizes (sensitivity+specificity)/2
4  MaxKappa   maximizes Kappa
5  MaxPCC     maximizes PCC (percent correctly classified)
6  PredPrev=Obs predicted prevalence=observed prevalence
7  ObsPrev    threshold=observed prevalence
8  MeanProb   mean predicted probability
9  MinROCdist minimizes distance between ROC plot and (0,1)
10 ReqSens    user defined required sensitivity
11 ReqSpec    user defined required specificity

req.sens      a value between zero and one giving the user defined required sensitivity. Only used if opt.thresholds = TRUE. Note that req.sens = (1-maximum allowable errors for points with positive observations).
req.spec      a value between zero and one giving the user defined required specificity. Only used if opt.thresholds = TRUE. Note that req.sens = (1- maximum allowable errors for points with negative observations).
obs.prev      observed prevalence for opt.method = "PredPrev=Obs" and "ObsPrev". Defaults to observed prevalence from DATA.
smoothing  smoothing factor for maximizing/minimizing. Only used if opt.thresholds = TRUE. Instead of find the threshold that gives the max/min value, function will average the thresholds of the given number of max/min values.

add.legend  a logical indicating if a legend for presence/absence should be added to plot

legend.text  a two item vector of text for presence/absence legend. Defaults to "present" and "absent".

legend.cex  cex for presence/absence legend

add.opt.legend  logical indicating if a legend for optimal threshold criteria should be included on the plot

opt.legend.text  a vector of text for optimimal threshold criteria legend. Defaults to text corresponding to 'opt.methods'.

opt.legend.cex  cex for optimization criteria legend

pch  plotting "character", i.e., symbol to use for the thresholds specified in mark. pch can either be a single character or an integer code for one of a set of graphics symbols. See help(points) for details.

FPC  False Positive Costs, or for C/B ratio C = 'net costs of treating nondiseased individuals'.

FNC  False Negative Costs, or for C/B ratio B = 'net benefits of treating diseased individuals'.

Details

When examining a Presence/Absence histogram to evaluate model quality, a good model will produce a clear separation of 'present' and 'absent' with little overlap in any bars.

The truncate.tallest argument is useful when one bar (often the bar for predicted probability of zero) is much larger than all the other bars. If truncate.tallest = TRUE, the tallest bar is truncated to slightly taller than the next highest bar, and the actual count is plotted above the bar. The truncated bar is also crosshatched to avoid confusion by making it more obviously different from the other bars.

if optimal.thresholds = TRUE the function will find optimal thresholds by several methods and plot them along the X axis. See optimal.thresholds for more details on the optimization methods, and on the arguments ReqSens, ReqSpec, obs.prev, smoothing, FPC, and FNC.

Note: if too many methods are included in opt.methods, the graph will get very crowded.

Value

creates a graphical plot

Author(s)

Elizabeth Freeman <eafreeman@fs.fed.us>

See Also

optimal.thresholds, presence.absence.summary
Examples

```r
## EXAMPLE 1 - Comparing three models
par(mfrow=c(1,3))
for(i in 1:3){
  presence.absence.hist( SIM3DATA,
    which.model=i,
    na.rm=TRUE,
    model.names=c("Model 1","Model 2","Model 3"),
    N.bars=10,
    truncate.tallest=FALSE,
    opt.thresholds=TRUE,
    opt.methods=c("Default","Sens=Spec","MaxKappa"))
}

## EXAMPLE 2 - Effect of 'truncate.tallest' argument
par(mfrow=c(1,2))
presence.absence.hist( SIM3DATA,
  which.model=1,
  model.names=c("Model 1","Model 2","Model 3"),
  N.bars=10,
  truncate.tallest=FALSE,
  main="truncate.tallest=FALSE")
presence.absence.hist( SIM3DATA,
  which.model=1,
  model.names=c("Model 1","Model 2","Model 3"),
  N.bars=10,
  truncate.tallest=TRUE,
  main="truncate.tallest=TRUE")
```

Description

`presence.absence.simulation` simulates presence/absence data as one set of observed values, and one or more prediction models. First, observed values are generated as a binomial distribution, then for each model two beta distributions are used to generate predicted values, one beta distribution for the data points where the simulated observed value is present, and a second for points where it is absent.

Usage

```r
presence.absence.simulation(n, prevalence, N.models = 1,
  shape1.absent, shape2.absent, shape1.present, shape2.present)
```
Arguments

- **n**: number of plots (i.e. rows) in simulated dataset
- **prevalence**: probability species is present for binomial observed values
- **N.models**: number of models to simulate predictions for
- **shape1.absent**: first parameter for beta distribution for plots where observed value is absent
- **shape2.absent**: second parameter for beta distribution for plots where observed value is absent
- **shape1.present**: first parameter for beta distribution for plots where observed value is present
- **shape2.present**: second parameter for beta distribution for plots where observed value is present

Details

`presence.absence.simulation` will generate predicted probabilities for one or more models. If **N.models** = 1, then shape parameters should be of length 1. If **N.models** > 1, then shape parameters can be either length 1 or vectors of length **N.models**.

The beta distribution is extremely flexible and is capable of generating data with unrealistic behavior. The following rules of thumb will help generate realistic datasets:

- The mean of the beta distribution equals shape1/(shape1+shape2). To get reasonable predictions (e.g. better than random), the mean for the plots where the observed value is present should be higher than that for the plots where the species is absent:
  \[ \text{mean(present)} > \text{mean(absent)} \]
- The overall mean probability should be approximately equal to the prevalence. In other words:
  \[ \text{prevalence} \times \text{mean(present)} + (1 - \text{prevalence}) \times \text{mean(absent)} = \text{prevalence} \]

Value

`presence.absence.simulation` returns a dataframe where:

- column 1: plotID - plot ID numbers
- column 2: Observed - 0/1 values
- column 3: Predicted 1 - predicted probabilities for model 1
- column 4: Predicted 2 - predicted probabilities for model 2, etc...

Author(s)

Elizabeth Freeman <eafreeman@fs.fed.us>

Examples

#### EXAMPLE 1 ####

```r
### a graph illustrating effect of shape parameters on beta distribution
set.seed(666)
shapes<-c(1,2,5,10,20)
par(mfrow=c(5,5),mar=c(2,2,2,2),oma=c(0,3,3,0))
```
for(i in 1:5){
  for(j in 1:5){
    SIMDATA<-presence.absence.simulation(n=1000,
      prevalence=1,
      N.models=1,
      shape1.absent=1,
      shape2.absent=1,
      shape1.present=shapes[i],
      shape2.present=shapes[j])
    }
  }
}

### EXAMPLE 2 ###
### generate observed data along with 3 sets of model predictions
### for models of varying predictive ability.
### Note: This is the code used to generate sample dataset SIM3DATA.
set.seed(666)
SIM3DATA<-presence.absence.simulation(n=1000,
  prevalence=.2,
  N.models=3,
  shape1.absent=c(1,1,1),
  shape2.absent=c(14,7,5),
  shape1.present=c(6,2,1),
  shape2.present=c(2,2,2))
Arguments

| DATA | a matrix or dataframe of observed and predicted values where each row represents one plot and where columns are: |
DATA[,1]  plot ID
DATA[,2]  observed values
DATA[,3]  predicted probabilities from first model
DATA[,4]  predicted probabilities from second model, etc...

threshold  cutoff values between zero and one used for translating predicted probabilities into 0/1 values, defaults to 0.5. It can be a single value between zero and one, a vector of values between zero and one, or a positive integer representing the number of evenly spaced thresholds to calculate.

find.auc  a logical indicating if area under the curve should be calculated

which.model  a number indicating which model from DATA should be used

na.rm  a logical indicating whether missing values should be removed

main  an overall title for the plot

model.names  a vector of the names of each model included in DATA to be used in the legend box

alpha  alpha value for confidence intervals for calibration.plot

N.bins  integer giving number of bins for predicted probabilities for calibration.plot

N.bars  number of bars in histogram

truncate.tallest  a logical indicating if the tallest bar should be truncated to fit for presence.absence.hist

opt.thresholds  a logical indicating whether the optimal thresholds should be calculated and plotted

opt.methods  what methods should be used to optimize thresholds. Argument can be given either as a vector of method names or method numbers. Possible values are:

1 Default  threshold=0.5
2 Sens=Spec  sensitivity=specificity
3 MaxSens+Spec  maximizes (sensitivity+specificity)/2
4 MaxKappa  maximizes Kappa
5 MaxPCC  maximizes PCC (percent correctly classified)
6 PredPrev=Obs  predicted prevalence=observed prevalence
7 ObsPrev  threshold=observed prevalence
8 MeanProb  mean predicted probability
9 MinROCDist  minimizes distance between ROC plot and (0,1)
10 ReqSens  user defined required sensitivity
11 ReqSpec  user defined required specificity

req.sens  a value between zero and one giving the user defined required sensitivity. Only used if opt.thresholds = TRUE. Note that req.sens = (1-maximum allowable errors for points with positive observations).

req.spec  a value between zero and one giving the user defined required specificity. Only used if opt.thresholds = TRUE. Note that req.spec = (1- maximum allowable errors for points with negative observations).

obs.prev  observed prevalence for opt.method = "PredPrev=Obs" and "ObsPrev". De-
presence.absence.summary

presence.absence.summary produces a set of summary plots for a single model, along with calculating AUC and optimal thresholds. presence.absence.summary is not quite as flexible as the individual plot functions, as some arguments are preset so that the plots will be comparable, but the remaining arguments have the same meaning. See the individual plot functions error.threshold.plot, auc.roc.plot, calibration.plot, and presence.absence.hist for further details.

Value

creates a graphical plot

Author(s)

Elizabeth Freeman <eafreeman@fs.fed.us>

See Also

optimal.thresholds, error.threshold.plot, auc.roc.plot, calibration.plot, presence.absence.hist

Examples

data(SIM3DATA)

presence.absence.summary(SIM3DATA)
roc.plot.calculate

**ROC Plot Calculations**

**Description**


**Usage**

roc.plot.calculate(DATA, threshold = 101, which.model = 1, na.rm = FALSE)

**Arguments**

- **DATA**
  - a matrix or dataframe of observed and predicted values where each row represents one plot and where columns are:
    - `DATA[,1]`: plot ID, text
    - `DATA[,2]`: observed values, zero-one values
    - `DATA[,3]`: predicted probabilities from first model, numeric (between 0 and 1)
    - `DATA[,4]`: predicted probabilities from second model, etc...

- **threshold**
  - cutoff values between zero and one used for translating predicted probabilities into 0/1 values, defaults to 0.5. It can be a single value between zero and one, a vector of values between zero and one, or a positive integer representing the
roc.plot.calculate

number of evenly spaced thresholds to calculate.

which.model a number indicating which model from DATA should be used

na.rm a logical indicating whether missing values should be removed

Details

roc.plot.calculate is a streamlined version of presence.absence.accuracy designed specifically to compute the accuracy measures needed to produce a ROC plot. roc.plot.calculate is less versatile, but more efficient than presence.absence.accuracy.

Unlike presence.absence.accuracy, roc.plot.calculate will only work for a single set of model predictions. Therefore either DATA can contain only one model prediction, or which.model must be used to indicate a single model prediction from DATA. By default, if DATA contains more than one model prediction, and which.model is not specified, roc.plot.calculate will use the first model prediction (e.g. DATA[,3]).

roc.plot.calculate was written as a sub-function for the plotting functions (i.e. error.threshold.plot, auc.roc.plot, but can be used on its own to produce a simple table of how the accuracy measures vary with choice of threshold.

To produce attractive plots requires a large number of thresholds. The default value of threshold = 101 is a good compromise between speed and resolution.

Value

Returns a dataframe where:

[.1] threshold thresholds used for each row in the table
[.2] PCC percent correctly classified
[.3] sensitivity
[.4] specificity
[.5] Kappa

Author(s)

Elizabeth Freeman <eafreeman@fs.fed.us>

See Also

cmx, pcc, sensitivity, specificity, Kappa, presence.absence.accuracy, error.threshold.plot, auc.roc.plot

Examples

data(SIM3DATA)

roc.plot.calculate(SIM3DATA, which.model=2)
sensitivity

Description

sensitivity calculates the sensitivity and (optionally) the associated standard deviation from a confusion matrix.

Usage

```r
sensitivity(CMX, st.dev = TRUE)
```

Arguments

- `CMX`: a confusion matrix - output from `cmx`
- `st.dev`: a logical indicating whether the associated standard deviation should be calculated

Details

Sensitivity is the proportion of observed positives correctly predicted, and reflects a model’s ability to predict a presence given that a species actually occurs at a location.

Value

- if `st.dev = FALSE`, returns: sensitivity.
- if `st.dev = TRUE`, returns a dataframe where:

  ```
  [1,1]  sensitivity
  [1,2]  sensitivity.sd  standard deviation of sensitivity
  ```

Author(s)

Elizabeth Freeman <eafreeman@fs.fed.us>

See Also

`cmx`, `pcc`, `specificity`, `Kappa`, `auc`

Examples

```r
data(SIM3DATA)
sensitivity(cmx(SIM3DATA))
sensitivity(cmx(SIM3DATA), st.dev=FALSE)
```
**SIM3DATA**  
*Simulated Presence-Absence Data*

**Description**

This data set was generated using the `presence.absence.simulation()` function. It consists of plotID, observed presence-absence values, and the simulated probability predictions of three different models, for 1000 plots.

**Usage**

```r
data(SIM3DATA)
```

**Format**

A data frame with 1000 observations on the following 5 variables.

- **plotID** a numeric vector
- **Observed** a numeric vector of 0-1 values
- **Predicted1** a numeric vector of predicted probabilities
- **Predicted2** a numeric vector of predicted probabilities
- **Predicted3** a numeric vector of predicted probabilities

**Source**

simulated with the `presence.absence.simulation` function. See Example 2 from `presence.absence.simulation` help file for more details.

**Examples**

```r
data(SIM3DATA)
```

---

**SPDATA**  
*Species Presence/Absence Data*

**Description**

This data set has Presence/Absence predictions for 13 species at 386 forested locations. It consists of species, observed presence-absence values, and the probability predictions of three different models.

**Usage**

```r
data(SPDATA)
```
**specificity**

**Format**
A data frame with 386 observations for 13 species with the following 5 variables.

- `SPECIES` a character vector of species codes
- `OBSERVED` a numeric vector of 0-1 values
- `GAM` a numeric vector of predicted probabilities
- `See5` a numeric vector of predicted probabilities
- `SGB` a numeric vector of predicted probabilities

**Source**
This dataset is from:

**Examples**
```r
data(SPDATA)
```

<table>
<thead>
<tr>
<th>specificity</th>
<th>Specificity</th>
</tr>
</thead>
</table>

**Description**
specificity calculates the specificity and (optionally) the associated standard deviation from a confusion matrix.

**Usage**
specificity(CMX, st.dev = TRUE)

**Arguments**
- `CMX` a confusion matrix - output from `cmx`
- `st.dev` a logical indicating whether the associated standard deviation should be calculated

**Details**
Specificity is the proportion of observed negatives correctly predicted, and reflects a model’s ability to predict an absence given that a species actually does not occur at a location.

**Value**
if `st.dev = FALSE`, returns: `specificity`.

if `st.dev = TRUE`, returns a dataframe where:
[1,1] specificity
[1,2] specificity.sd  standard deviation of specificity

Author(s)
Elizabeth Freeman <eafreeman@fs.fed.us>

See Also
cmx, pcc, sensitivity, Kappa, auc

Examples
data(SIM3DATA)
specificity(cmx(SIM3DATA))
specificity(cmx(SIM3DATA), st.dev=FALSE)

SPPREV Overall Prevalences for Species Presence/Absence Data

Description
This data set is summary prevalence for Presence/Absence data for 13 species from 1930 forested locations. SPPREV is the prevalence data from the full dataset (training and test data). Note that SPDATA is the model predictions from the test data subset (20 percent of total plots) of this original dataset, and therefore the species in SPDATA have slightly different prevalence than the overall prevalence given in SPPREV.

Usage
data(SPDATA)

Format
A data frame with data for 13 species. Dataframe consists of species names, number of plots where the species was present, and the overall prevalence for each species:

SPECIES  a character vector of species codes
NUMPLOTS  a numeric vector of plot counts
PREV  a numeric vector of prevalence

Source
This dataset is from:
Examples

data(SPPREV)
Index

* datagen
  presence.absence.simulation, 29

* datasets
  SIM3DATA, 38
  SPDATA, 38
  SPPREV, 40

* models
  auc, 4
  auc.roc.plot, 5
  calibration.plot, 9
  cmx, 11
  error.threshold.plot, 13
  Kappa, 17
  optimal.thresholds, 18
  pcc, 22
  predicted.prevalence, 23
  presence.absence.accuracy, 24
  presence.absence.hist, 26
  presence.absence.summary, 31
  roc.plot.calculate, 35
  sensitivity, 37
  specificity, 39

* package
  PresenceAbsence-package, 2

PresenceAbsence
  (PresenceAbsence-package), 2
PresenceAbsence-package
  (PresenceAbsence-package), 2

auc, 3, 4, 17, 22, 25, 37, 40
auc.roc.plot, 3, 5, 34, 36

calibration.plot, 3, 9, 34
cmx, 2, 3, 5, 11, 17, 22, 25, 36, 37, 40
error.threshold.plot, 3, 8, 13, 21, 34, 36
Kappa, 3, 5, 12, 17, 22, 25, 36, 37, 40
optimal.thresholds, 3, 8, 15, 16, 18, 28, 34

pcc, 2, 3, 5, 12, 17, 22, 25, 36, 37, 40
predicted.prevalence, 3, 23
presence.absence.accuracy, 3, 8, 16, 21, 24, 36
presence.absence.hist, 3, 11, 26, 34
presence.absence.simulation, 29, 38
presence.absence.summary, 3, 8, 11, 16, 21, 28, 31
PresenceAbsence-package, 2
roc.plot.calculate, 3, 8, 16, 21, 35

sensitivity, 3, 5, 12, 17, 22, 25, 36, 37, 40
SIM3DATA, 38
SPDATA, 38
specificity, 3, 5, 12, 17, 22, 25, 36, 37, 39
SPPREV, 40

42