Package ‘wevid’

May 11, 2026

Type Package

Title Weight of Evidence for Quantifying Performance of a Binary Classifier

Version 0.7.0

Date 2026-05-10

Description The distributions of the weight of evidence (log Bayes factor) favouring case over noncase status in a test dataset (or test folds generated by cross-validation) can be used to quantify the performance of a diagnostic test. This package can be used with any test dataset on which you have computed prior probabilities of case status, posterior probabilities of case status, and you have the observed case-control status. In comparison with the C-statistic (area under ROC curve), the expected weight of evidence (expected information for discrimination) has several advantages as a summary measure of predictive performance. To quantify how the predictor will behave as a risk stratifier, the quantiles of the distributions of weight of evidence in cases and controls can be calculated and plotted.

Depends R (>= 2.10)

License GPL-3

URL https://precmed.cphs.mvm.ed.ac.uk/pmckeigue/preprints/cstatistic.pdf

LazyData true

Imports ggplot2, pROC, reshape2, zoo

ByteCompile TRUE

Encoding UTF-8

RoxygenNote 7.3.3

NeedsCompilation no

Author Paul McKeigue [aut, cre],
Marco Colombo [ctb]

Maintainer Paul McKeigue <`paul.mckeigue@ed.ac.uk`>

Repository CRAN

Date/Publication 2026-05-11 06:50:02 UTC

Contents

`wevid-package` *Quantifying performance of a diagnostic test using the sampling distribution of the weight of evidence favouring case over noncase status*

Description

This package provides functions for quantifying the performance of a diagnostic test (or any other binary classifier) by calculating and plotting the distributions in cases and noncases of the weight of evidence favouring case over noncase status.

Details

To use it, you should have computed on a test dataset (or on test folds used for cross-validation:

1. The prior probability of case status (this may be just the frequency of cases in the training data.

2. The posterior probability of case status (using the model learned on the training data to predict on the test data)

3. The observed case status (coded as 0=noncase, 1=case).

Author(s)

Paul McKeigue `<paul.mckeigue@ed.ac.uk>`

Citation for the statistical methods used in this package: McKeigue P. Quantifying performance of a diagnostic test as the expected information for discrimination: relation to the C-statistic. Statistical Methods for Medical Research 2018, in press.

`auroc.model` *Compute area under the ROC curve according to model-based densities*

Description

Compute area under the ROC curve according to model-based densities

Usage

```
auroc.model(densities)
```

Arguments

`densities`	Adjusted densities computed by `Wdensities.fromraw`.

Value

The area under the model-based ROC curve computed from the densities of the weight of evidence in cases and noncases. This model-based ROC curve is always concave (if the densities have been adjusted to make them mathematically consistent).

See Also

plotroc

Examples

```
data("cleveland") # load example dataset
W <- with(cleveland, weightsofevidence(posterior.p, prior.p))
densities.unadj <- Wdensities.unadjusted(cleveland$y, W)
densities.adj <- Wdensities.fromraw(densities.unadj)
auroc.model(densities.adj)
```

`cleveland` *Example dataset based on cross-validated prediction of outcome in the Cleveland Heart Study*

Description

Example dataset based on cross-validated prediction of outcome in the Cleveland Heart Study

Usage

```
cleveland
```

Format

A data frame with 297 rows and three variables

prior.p	prior probabilities of case status
posterior.p	posterior probabilites of case status
y	case-control status

`lambda.model` *Compute the expected information for discrimination (expected weight of evidence) from the model-based densities*

Description

Compute the expected information for discrimination (expected weight of evidence) from the model-based densities

Usage

```
lambda.model(densities)
```

Arguments

`densities`	Adjusted densities computed by `Wdensities.fromraw`.

Value

The model-based expected information for discrimination.

Examples

```
data("cleveland") # load example dataset
W <- with(cleveland, weightsofevidence(posterior.p, prior.p))
densities.unadj <- Wdensities.unadjusted(cleveland$y, W)
densities.adj <- Wdensities.fromraw(densities.unadj)
lambda.model(densities.adj)
```

`means.densities` *Means of densities in cases and controls*

Description

Means of densities in cases and controls

Usage

```
means.densities(densities)
```

Arguments

`densities`	Adjusted densities computed by `Wdensities.fromraw`.

Value

numeric vector of length 2: mean densities in controls and in cases.

See Also

`lambda.model`

`plotcumfreqs` *Plot the cumulative frequency distributions in cases and in controls*

Description

Plot the cumulative frequency distributions in cases and in controls

Usage

```
plotcumfreqs(densities)
```

Arguments

`densities`	Adjusted densities computed by `Wdensities.fromraw`.

Examples

```
data("cleveland") # load example dataset
W <- with(cleveland, weightsofevidence(posterior.p, prior.p))
densities.unadj <- Wdensities.unadjusted(cleveland$y, W)
densities.adj <- Wdensities.fromraw(densities.unadj)
plotcumfreqs(densities.adj)
```

`plotroc` *Plot crude and model-based ROC curves*

Description

Plot crude and model-based ROC curves

Usage

```
plotroc(densities, y, W)
```

Arguments

`densities`	Adjusted densities computed by `Wdensities.fromraw`.
`y`	Binary outcome label (0 for controls, 1 for cases).
`W`	Weight of evidence (natural logs).

Value

ggplot of crude and model-based ROC curves

Examples

```
data("cleveland") # load example dataset
W <- with(cleveland, weightsofevidence(posterior.p, prior.p))
densities.unadj <- Wdensities.unadjusted(cleveland$y, W)
densities.adj <- Wdensities.fromraw(densities.unadj)
plotroc(densities.adj, cleveland$y, W)
```

`plotW` *plot log case/control density ratio against weight of evidence as a check that the densities are mathematically consistent*

Description

plot log case/control density ratio against weight of evidence as a check that the densities are mathematically consistent

Usage

```
plotW(densities, W)
```

Arguments

`densities`	Adjusted densities computed by `Wdensities.fromraw`.
`W`	Weight of evidence. (natural logs)

Value

ggpplot of natural log case/control density ratio against weight of evidence (should be a straight line of gradient 1 passing through the origin)

Examples

```
data("cleveland") # load example dataset
W <- with(cleveland, weightsofevidence(posterior.p, prior.p))
densities.unadj <- Wdensities.unadjusted(cleveland$y, W)
densities.adj <- Wdensities.fromraw(densities.unadj)
plotW(densities.adj, W)
```

`plotWdists` *Plot the distribution of the weight of evidence in cases and in controls*

Description

Plot the distribution of the weight of evidence in cases and in controls

Usage

```
plotWdists(
  Wdensities.unadj,
  Wdensities.adj,
  mask = NULL,
  distlabels = c("Crude", "Adjusted")
)
```

Arguments

`Wdensities.unadj`	Unadjusted densities computed by `Wdensities.unadjusted`.
`Wdensities.adj`	Adjusted densities computed by `Wdensities.fromraw`.
`mask`	if not null, breaks y axis to show more detail of lower end
`distlabels`	Character vector of length 2

Examples

```
data("cleveland") # load example dataset
W <- with(cleveland, weightsofevidence(posterior.p, prior.p))
densities.unadj <- Wdensities.unadjusted(cleveland$y, W)
densities.adj <- Wdensities.fromraw(densities.unadj)
plotWdists(densities.unadj, densities.adj)
```

`prop.belowthreshold` *Proportions of cases and controls below a given threshold of W (natural logs)*

Description

Proportions of cases and controls below a given threshold of W (natural logs)

Usage

```
prop.belowthreshold(densities, w.threshold)
```

Arguments

`densities`	Adjusted densities computed by `Wdensities.fromraw`.
`w.threshold`	Threshold value of weight of evidence (natural logs).

Value

numeric vector of length 2: proportions of controls and cases with weight of evidence below the given threshold.

Examples

```
data("cleveland") # load example dataset
W <- with(cleveland, weightsofevidence(posterior.p, prior.p))
densities.unadj <- Wdensities.unadjusted(cleveland$y, W)
densities.adj <- Wdensities.fromraw(densities.unadj)
w.threshold <- log(4) # threshold Bayes factor of 4
prop.belowthreshold(densities.adj, w.threshold)
```

`Wdensities.fromraw` *Adjust the crude densities of weights of evidence in cases and controls to make them mathematically consistent*

Description

Adjust the crude densities of weights of evidence in cases and controls to make them mathematically consistent

Usage

```
Wdensities.fromraw(densities)
```

Arguments

`densities`	Unadjusted densities computed by `Wdensities.unadjusted`.

Examples

```
data("cleveland") # load example dataset
W <- with(cleveland, weightsofevidence(posterior.p, prior.p))
densities.unadj <- Wdensities.unadjusted(cleveland$y, W)
densities.adj <- Wdensities.fromraw(densities.unadj)
plotWdists(densities.unadj, densities.adj)
```

`Wdensities.mix` *Compute smoothed densities for a spike-slab mixture distribution*

Description

Compute smoothed densities for a spike-slab mixture distribution

Usage

```
Wdensities.mix(y, W, in.spike, range.xseq = c(-25, 25), x.stepsize = 0.01)
```

Arguments

`y`	Binary outcome label (0 for controls, 1 for cases).
`W`	Weight of evidence.
`in.spike`	logical vector same length as y, TRUE if in spike component, FALSE otherwise. Typically used where high proportion of values of the predictor are zero
`range.xseq`	Range of points where the curves should be sampled.
`x.stepsize`	Distance between each point.

`Wdensities.unadjusted` *Calculate the unadjusted smoothed densities of W in cases and in controls*

Description

Calculate the unadjusted smoothed densities of W in cases and in controls

Usage

```
Wdensities.unadjusted(
  y,
  W,
  range.xseq = c(-25, 25),
  x.stepsize = 0.01,
  adjust.bw = 1
)
```

Arguments

`y`	Binary outcome label (0 for controls, 1 for cases).
`W`	Weight of evidence.
`range.xseq`	Range of points where the curves should be sampled.
`x.stepsize`	Distance between each point.
`adjust.bw`	Bandwidth adjustment.

Value

unadjusted density object

`weightsofevidence` *Calculate weights of evidence in natural log units*

Description

Calculate weights of evidence in natural log units

Usage

```
weightsofevidence(posterior.p, prior.p)
```

Arguments

`posterior.p`	Vector of posterior probabilities generated by using model to predict on test data
`prior.p`	Prior probabilities on test data.

Value

The weight of evidence in nats for each observation.

Examples

```
data("cleveland") # load example dataset
W <- with(cleveland, weightsofevidence(posterior.p, prior.p))
densities.unadj <- Wdensities.unadjusted(cleveland$y, W)
densities.adj <- Wdensities.fromraw(densities.unadj)
plotWdists(densities.unadj, densities.adj)
```

`wtrue.results` *Summary evaluation of predictive performance*

Description

Summary evaluation of predictive performance

Usage

```
wtrue.results(studyname, y, posterior.p, prior.p)
```

Arguments

`studyname`	Name of the study.
`y`	Binary outcome label (0 for controls, 1 for cases).
`posterior.p`	Vector of posterior probabilities generated by a logistic regression model.
`prior.p`	Vector of prior probabilities.

Value

A dataframe listing some metrics of predictive performance.

Examples

```
data("cleveland") # load example dataset
with(cleveland,
     wtrue.results("cleveland", y, posterior.p, prior.p))
```

Index