Package ‘c2c’

July 23, 2017

Type Package

Title Compare Two Classifications or Clustering Solutions of Varying Structure

Version 0.1.0

Maintainer Mitchell Lyons <mitchell.lyons@gmail.com>

Description Compare two classifications or clustering solutions that may or may not have the same number of classes, and that might have hard or soft (fuzzy, probabilistic) membership. Calculate various metrics to assess how the clusters compare to each other. The calculations are simple, but provide a handy tool for users unfamiliar with matrix multiplication. This package is not geared towards traditional accuracy assessment for classification/mapping applications - the motivating use case is for comparing a probabilistic clustering solution to a set of reference or existing class labels that could have any number of classes (that is, without having to degrade the probabilistic clustering to hard classes).

Depends R (>= 3.1.0)

URL https://github.com/mitchest/c2c/

BugReports https://github.com/mitchest/c2c/issues

License GPL-3

Encoding UTF-8

LazyData true

Suggests testthat, knitr, rmarkdown, e1071

RoxygenNote 6.0.1

VignetteBuilder knitr

NeedsCompilation no

Author Mitchell Lyons [aut, cre]

Repository CRAN

Date/Publication 2017-07-23 17:50:40 UTC
\textbf{R topics documented:}

\begin{itemize}
\item calculate\_clustering\_metrics \hspace{1cm} 2
\item class\_entropy \hspace{1cm} 3
\item class\_purity \hspace{1cm} 4
\item get\_conf\_mat \hspace{1cm} 5
\item get\_hard \hspace{1cm} 6
\item labels\_to\_matrix \hspace{1cm} 7
\item overall\_entropy \hspace{1cm} 8
\item overall\_purity \hspace{1cm} 8
\item percentage\_agreement \hspace{1cm} 9
\end{itemize}

\textbf{Index}

\begin{itemize}
\item calculate\_clustering\_metrics \hspace{1cm} 10
\end{itemize}

\underline{calculate\_clustering\_metrics}

\textit{Calculate clustering metrics for a confusion matrix}

\textbf{Description}

Calculate a range of clustering metrics on a confusion confusion matrix, usually from \texttt{get\_conf\_mat}.

\textbf{Usage}

\begin{verbatim}
calculate_clustering_metrics(conf_mat)
\end{verbatim}

\textbf{Arguments}

\begin{verbatim}
conf_mat \hspace{1cm} a confusion matrix, as produced by \texttt{get\_conf\_mat}, or otherwise a confusion matrix of the same form.
\end{verbatim}

\textbf{Details}

Entropy calculated via \texttt{overall\_entropy} and \texttt{class\_entropy}, purity calculated via \texttt{overall\_purity} and \texttt{class\_purity}, percentage agreement calculated via \texttt{percentage\_agreement} (only for confusion matrices of equal dimensions and matching class order)

\textbf{Value}

A list containing the metrics that can be calculated, see details.

\textbf{Author(s)}

Mitchell Lyons

\textbf{References}

class_entropy

See Also

get_conf_mat, labels_to_matrix, get_hard

Examples

# meaningless data, but you get the idea

# compare two soft classifications
my_soft_mat1 <- matrix(runif(50,0,1), nrow = 10, ncol = 5)
my_soft_mat2 <- matrix(runif(30,0,1), nrow = 10, ncol = 3)
# make the confusion matrix and calculate stats
conf_mat <- get_conf_mat(my_soft_mat1, my_soft_mat2)
conf_mat; calculate_clustering_metrics(conf_mat)

# compare a soft classification to a vector of hard labels
my_labels <- rep(c("a","b","c"), length.out = 10)
# utilising labels_to_matrix(my_labels)
conf_mat <- get_conf_mat(my_soft_mat1, my_labels)
conf_mat; calculate_clustering_metrics(conf_mat)

# make one of the soft matrices hard
# utilising get_hard(my_soft_mat2)
conf_mat <- get_conf_mat(my_soft_mat1, my_soft_mat2, make.B.hard = TRUE)
conf_mat; calculate_clustering_metrics(conf_mat)

# two classifications with same number of classes, enables percentage agreement
conf_mat <- get_conf_mat(my_soft_mat1, my_soft_mat1)
conf_mat; calculate_clustering_metrics(conf_mat)

_________________________________________________________________

class_entropy

Calculate cluster entropy per class

Description

Used to calculate cluster entropy from a confusion matrix, for each class (i.e. each row and column of the confusion matrix).

Usage

class_entropy(conf_mat)

Arguments

conf_mat A confusion matrix from get_conf_mat or otherwise (ideally a matrix, although data frames will probably work)
class_purity

Details
Metrics per class are useful when you are comparing two classifications with different numbers of classes, when an overall measure might not be useful or sensible. Entropy as defined in Manning (2008).

Value
A data frame with two columns, the first corresponding to the confusion matrix rows, the second corresponding to the confusion matrix columns.

References

class_purity

Calculate cluster purity per class

Description
Used to calculate cluster purity from a confusion matrix, for each class (i.e. each row and column of the confusion matrix).

Usage
class_purity(conf_mat)

Arguments
conf_mat A confusion matrix from get_conf_mat or otherwise (ideally a matrix, although data frames will probably work)

Details
Metrics per class are useful when you are comparing two classifications with different numbers of classes, when an overall measure might not be useful or sensible. Purity as defined in Manning (2008).

Value
A data frame with two columns, the first corresponding to the confusion matrix rows, the second corresponding to the confusion matrix columns.

References
**get_conf_mat**

Generate a confusion matrix from two classifications/clustering solutions.

### Description

`get_conf_mat` takes two classifications or clustering solutions and creates a confusion matrix representing the number of shared sites between them.

### Usage

```r
get_conf_mat(A, B, make.A.hard = F, make.B.hard = F)
```

### Arguments

- **A**: A matrix or data.frame (or something that can be coerced to a matrix) of class membership or a vector of class labels (character or factor).
- **B**: A matrix or data.frame (or something that can be coerced to a matrix) or class membership or a vector of class labels (character or factor).
- **make.A.hard**: logical (defaults to FALSE). If TRUE, and if `A=` is a matrix of soft membership, it will be degraded to a hard binary matrix, taking the highest value, breaking ties at random.
- **make.B.hard**: logical (defaults to FALSE). If TRUE, and if `B=` is a matrix of soft membership, it will be degraded to a hard binary matrix, taking the highest value, breaking ties at random.

### Details

Takes inputs `A` and `B` (converting labels to matrices if required) and combines them via `(A^T B)`. Soft classifications will necessarily be matrices. Hard classifications can be given as a binary matrix of membership or a vector of labels. For matrix inputs, rows should represent individual sites, observations, cases etc., and columns should represent classes. For class label inputs, the vector should be ordered similarly by site, observation, case etc; they will be converted to a binary matrix (see `labels_to_matrix`). Classes from matrix `A` are represented by rows of the output, and classes from matrix `B` are represented by the columns. Class names inherited from `names()` or `colnames()` - if at least one of the inputs has names, interpretation will be much easier. Ties in membership probability are broken at random - if you don’t want this to happen, suggest you break the tie manually before proceeding.

### Value

A confusion matrix

### Author(s)

Mitchell Lyons
get_hard

Decompose soft (fuzzy, probabilistic) membership to hard binary matrix

Description

Used in get_conf_mat but might be useful separately

Usage

get_hard(x)
labels_to_matrix

Arguments

x  A matrix or data frame (or something coercible to a matrix) containing memberships - rows are sites (observations, cases etc.) columns are classes

Value

Binary matrix of class membership. Class names inherited from names() or colnames().

Examples

my_mat <- matrix(runif(20,0,1), nrow = T)
geet_hard(my_mat)

labels_to_matrix

Make a vector of class labels into a hard binary matrix

Description

Used in get_conf_mat but might be useful separately

Usage

labels_to_matrix(x)

Arguments

x  Character or factor vector of class labels

Value

Binary matrix of class membership.

Examples

my_labels <- rep(c("a","b","c","d"), 5)
labels_to_matrix(my_labels)
overall_entropy  Calculate overall cluster entropy

Description

Used to calculate overall cluster entropy from a confusion matrix.

Usage

overall_entropy(conf_mat)

Arguments

cnf_mat  A confusion matrix from get_conf_mat or otherwise (ideally a matrix, although data frames will probably work)

Value

A scaler, cluster entropy as defined in Manning (2008)

References


overall_purity  Calculate overall cluster purity

Description

Used to calculate overall cluster purity from a confusion matrix.

Usage

overall_purity(conf_mat)

Arguments

cnf_mat  A confusion matrix from get_conf_mat or otherwise (ideally a matrix, although data frames will probably work)

Value

A scaler, cluster purity as defined in Manning (2008)
percentage_agreement

References


percentage_agreement  Calculate overall percentage agreement

Description

Used to calculate overall percentage agreement for a confusion matrix - the confusion matrix must have equal dimensions and the diagonal must represent 'matching' class pairs (percentage agreement does not make sense otherwise)

Usage

percentage_agreement(conf_mat)

Arguments

- conf_mat: A confusion matrix from get_conf_mat or otherwise (ideally a matrix, although data frames will probably work)

Value

A scaler, percentage agreement (sometime referred to as overall accuracy)
Index

*Topic accuracy
   calculate_clustering_metrics, 2
*Topic cluster
   calculate_clustering_metrics, 2
*Topic confusion
   calculate_clustering_metrics, 2
   get_conf_mat, 5
*Topic matrix,
   calculate_clustering_metrics, 2
*Topic matrix
   get_conf_mat, 5
*Topic metrics,
   calculate_clustering_metrics, 2
   calculate_clustering_metrics, 2, 6
   class_entropy, 2, 3
   class_purity, 2, 4
   get_conf_mat, 2–4, 5, 6–9
   get_hard, 3, 6, 6
   labels_to_matrix, 3, 5, 6, 7
   overall_entropy, 2, 8
   overall_purity, 2, 8
   percentage_agreement, 2, 9