Package ‘codacore’

January 7, 2022

Title Learning Sparse Log-Ratios for Compositional Data

Version 0.0.3

Description In the context of high-throughput genetic data, CoDaCoRe identifies a set of sparse biomarkers that are predictive of a response variable of interest (Gordon-Rodriguez et al., 2021) <doi:10.1093/bioinformatics/btab645>. More generally, CoDaCoRe can be applied to any regression problem where the independent variable is Compositional (CoDa), to derive a set of scale-invariant log-ratios (ILR or SLR) that are maximally associated to a dependent variable.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Depends R (>= 3.6.0)

Imports tensorflow (>= 2.1), keras (>= 2.3), pROC (>= 1.17), R6 (>= 2.5), gtools(>= 3.8)

SystemRequirements TensorFlow (https://www.tensorflow.org/)

Suggests zCompositions, testthat (>= 2.1.0), knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Elliott Gordon-Rodriguez [aut, cre],
Thomas Quinn [aut]

Maintainer Elliott Gordon-Rodriguez <eg2912@columbia.edu>

Repository CRAN

Date/Publication 2022-01-07 10:10:02 UTC

R topics documented:

`activeInputs.codacore` *activeInputs*

Description

activeInputs

Usage

```
activeInputs.codacore(cdcr)
```

Arguments

`cdcr`	A codacore object.

Value

The covariates included in the log-ratios

codacore *codacore*

Description

This function implements the codacore algorithm described by Gordon-Rodriguez et al. 2021 (https://doi.org/10.1101/2021.02.11.430695).

Usage

```
codacore(
  x,
  y,
  logRatioType = "balances",
  objective = NULL,
  lambda = 1,
  offset = NULL,
  shrinkage = 1,
  maxBaseLearners = 5,
  optParams = list(),
  cvParams = list(),
  verbose = FALSE,
  overlap = TRUE,
  fast = TRUE
)
```

Arguments

`x` A data.frame or matrix of the compositional predictor variables.

`y` A data.frame, matrix or vector of the response.

`logRatioType` A string indicating whether to use "balances" or "amalgamations". Also accepts "balance", "B", "ILR", or "amalgam", "A", "SLR". Note that the current implementation for balances is not strictly an ILR, but rather just a collection of balances (which are possibly non-orthogonal in the Aitchison sense).

`objective` A string indicating "binary classification" or "regression". By default, it is NULL and gets inferred from the values in y.

`lambda` A numeric. Corresponds to the "lambda-SE" rule. Sets the "regularization strength" used by the algorithm to decide how to harden the ratio. Larger numbers tend to yield fewer, more sparse ratios.

`offset` A numeric vector of the same length as y. Works similarly to the offset in a glm.

`shrinkage` A numeric. Shrinkage factor applied to each base learner. Defaults to 1.0, i.e., no shrinkage applied.

`maxBaseLearners`

An integer. The maximum number of log-ratios that the model will learn before stopping. Automatic stopping based on `seRule` may occur sooner.

`optParams` A list of named parameters for the optimization of the continuous relaxation. Empty by default. User can override as few or as many of our defaults as desired. Includes adaptiveLR (learning rate under adaptive training scheme), momentum (in the gradient-descent sense), epochs (number of gradient-descent epochs), batchSize (number of observations per minibatch, by default the entire dataset), and vanillaLR (the learning rate to be used if the user does *not* want to use the ’adaptiveLR’, to be used at the risk of optimization issues).

`cvParams` A list of named parameters for the "hardening" procedure using cross-validation. Includes numFolds (number of folds) and maxCutoffs (number of candidate cutoff values of ’c’ to be tested out during CV process).

`verbose` A boolean. Toggles whether to display intermediate steps.

`overlap` A boolean. Toggles whether successive log-ratios found by CoDaCoRe may contain repeated input variables. TRUE by default. Changing to FALSE implies that the log-ratios obtained by CoDaCoRe will become orthogonal in the Aitchison sense, analogously to the isometric-log-ratio transformation, while losing a small amount of model flexibility.

`fast` A boolean. Whether to run in fast or slow mode. TRUE by default. Running in slow mode will take ~x5 the computation time, but may help identify slightly more accurate log-ratios.

Value

A `codacore` object.

Examples

```
data("Crohn")
x <- Crohn[, -ncol(Crohn)]
y <- Crohn[, ncol(Crohn)]
x <- x + 1
model = codacore(x, y)
print(model)
plot(model)
```

Crohn *Microbiome composition related to Crohn‘s disease study*

Description

A dataset containing the number of counts of 48 different genera in a group of 975 samples (including 662 samples of patients with Crohn‘s disease and 313 controls). The `data.frame` is composed by 48 genera and a factor variable

Format

The `data.frame` is composed by 48 genera and a variable

genera The first 48 columns, from *g_Turicibacter* until *g_Bilophila* referred to different genera.

y a `factor` indicating if the sample corresponds to a case (*CD*) or a control (*no*).

References

https://qiita.ucsd.edu/

`FranzosaMetabolite` *Metabolite relative abundances (Franzosa et al., 2019)*

Description

A dataset containing the relative abundances of 7156 metabolites in a group of of 220 samples, together with an additional response variable indicating the corresponding Diagnosis.

Format

The `data.frame` is composed by metabolite data and Diagnosis

Metabolites TBD

Diagnosis Indicates if the sample was diagnosed with Crohn's disease (*CD*), ulcerative colitis (*UC*), or was a control (*Control*).

References

https://www.nature.com/articles/s41564-018-0306-4

`FranzosaMicrobiome` *Micriobiome relative abundances (Franzosa et al., 2019)*

Description

A dataset containing the relative abundances of 58 bacteria in a group of of 220 samples, together with an additional response variable indicating the corresponding Diagnosis.

Format

The `data.frame` is composed by microbiome data and Diagnosis

Microbiome The first 58 columns.

Diagnosis Indicates if the sample was diagnosed with Crohn's disease (*CD*), ulcerative colitis (*UC*), or was a control (*Control*).

References

https://www.nature.com/articles/s41564-018-0306-4

getDenominatorParts *getDenominatorParts*

Description

getDenominatorParts

Usage

```
getDenominatorParts(cdcr, baseLearnerIndex = 1)
```

Arguments

`cdcr`	A codacore object.
`baseLearnerIndex`	An integer indicating which of the (possibly multiple) log-ratios learned by codacore to be used.

Value

The covariates in the denominator of the selected log-ratio.

getLogRatios *getLogRatios*

Description

getLogRatios

Usage

```
getLogRatios(cdcr, x = NULL)
```

Arguments

`cdcr`	A codacore object
`x`	A set of (possibly unseen) compositional data. The covariates must be passed in the same order as for the original codacore() call.

Value

The learned log-ratio features, computed on input x.

`getNumeratorParts` *getNumeratorParts*

Description

getNumeratorParts

Usage

```
getNumeratorParts(cdcr, baseLearnerIndex = 1)
```

Arguments

`cdcr` A codacore object.

`baseLearnerIndex` An integer indicating which of the (possibly multiple) log-ratios learned by codacore to be used.

Value

The covariates in the numerator of the selected log-ratio.

`getSlopes` *getSlopes*

Description

getSlopes

Usage

```
getSlopes(cdcr)
```

Arguments

`cdcr` A codacore object

Value

The slopes (i.e., regression coefficients) for each log-ratio.

HIV *Microbiome, HIV infection and MSM factor*

Description

A dataset containing the number of counts of 60 different genera in a group of 155 samples (including HIV - infected and non - infected patients). The `data.frame` is composed by 60 genera and two variables.

Format

The `data.frame` is composed by 60 genera and 2 variables

genera The first 60 columns, from *g_Prevotella* until *o_NB1-n_g_unclassified* referred to different genera.

MSM a factor determining if the individual is `MSM` (*Men Sex with Men*) or not (`nonMSM`).

HIV_Status a factor specifying if the individual is infected (`Pos`) or not (`Neg`).

References

https://pubmed.ncbi.nlm.nih.gov/27077120/

plot.codacore *plot*

Description

Plots a summary of a fitted codacore model. Credit to the authors of the selbal package (Rivera-Pinto et al., 2018), from whose package these plots were inspired.

Usage

```
## S3 method for class 'codacore'
plot(x, index = 1, ...)
```

Arguments

`x`	A codacore object.
`index`	The index of the log-ratio to plot.
`...`	Not used.

`plotROC` *plotROC*

Description

plotROC

Usage

```
plotROC(cdcr)
```

Arguments

`cdcr`	A codacore object.

`predict.codacore` *predict*

Description

predict

Usage

```
## S3 method for class 'codacore'
predict(object, newx, asLogits = TRUE, numLogRatios = NA, ...)
```

Arguments

`object`	A codacore object.
`newx`	A set of inputs to our model.
`asLogits`	Whether to return outputs in logit space (as opposed to probability space). Should always be set to TRUE for regression with continuous outputs, but can be toggled for classification problems.
`numLogRatios`	How many predictive log-ratios to include in the prediction. By default, includes the effects of all log-ratios that were obtained during training. Setting this parameter to an integer k will restrict to using only the top k log-ratios in the model.
`...`	Not used.

`print.codacore` *print*

Description

print

Usage

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

Arguments

`x`	A codacore object.
`...`	Not used.

`sCD14` *Microbiome and sCD14 inflammation parameter*

Description

A dataset containing the number of counts of 60 different genera in a group of 151 samples (including HIV - infected and non - infected patients). The `data.frame` is composed by 60 genera and a numeric variable

Format

The `data.frame` is composed by 60 genera and a variable

genera The first 60 columns, from *g_Prevotella* until *o_NB1-n_g_unclassified* referred to different genera.

sCD14 a `numeric` variable with the value of the inflammation parameter sCD14 for each sample.

References

doi: 10.1016/j.ebiom.2016.01.032

simulateHTS *simulateHTS*

Description

This function simulates a set of (x, y) pairs. The covariates x are compositional, meaning they only carry relative information. The response y is a binary indicator. The rule linking x and y can be a balance or an amalgamation.

Usage

```
simulateHTS(n, p, outputType = "binary", logratio = "simple")
```

Arguments

n	Number of observations.
p	Number of covariates.
outputType	A string indicating ’binary’ or ’continuous’.
logratio	A string indicating ’simple’, ’balance’, or ’amalgamation’.

Value

A list containing a matrix of inputs and a vector of outputs

Index