Package 'tmsClassifier'

December 23, 2019

Type Package

Title Random Forest Classifiers for Transcranial Magnetic Stimulation Data

Version 0.1.0

Date 2019-12-20

Author Mario Grassi [aut], Fernando Palluzzi [aut, cre]

Maintainer Fernando Palluzzi <fernando.palluzzi@gmail.com>

Description

The tmsClassifier package predicts subject diagnosis (Frontotemporal Dementia, Alzheimer's Disease, Dementia with Lewy Bodies, or Healthy) based on Transcranial Magnetic Stimulation (TMS) data, through the integration of linear modeling and Random Forest Classifiers.

License GPL-3

Encoding UTF-8

LazyData TRUE

Depends R (>= 3.4.4)

Imports CMA (>= 1.36.0), impute (>= 1.52.0), methods (>= 3.4.4), randomForest (>= 4.6-14)

Suggests BiocStyle, knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 6.1.1

R topics documented:

`buildPredictor` *Build a RFC and test its performances*

Description

Build a RFC from TMS regression parameters and test its classification performances.

Usage

```
buildPredictor(data, status = 10, vset = NULL, n = 10000, m = 3)
```

Arguments

`data`	A matrix or data.frame containing subjects as rows and TMS parameters as columns. A further column specifying subject diagnosis can be given to validate the classifier.
`status`	Numeric value specifying subject diagnosis (default status = 10).
`vset`	An optional data.frame of the same format of the input data. This will be used as external validation set. The validation set can be generated from input data, using `tmsClassify`.
`n`	Number of bootstrap sampled trees of the RFC (default n = 10000). Increasing n improves RFC classification performance, but will sensibly increase computational time.
`m`	Number of random variables for each tree split (default m = 3). The suggested value is the squared root of the total number of variables (the default value is set to 3 since 9 TMS parameter are expected by the default classifier).

Value

An object of class `randomForest`. An optional object of class `performance` is returned, if vset is not NULL.

References

Liaw A, Wiener M. Classification and Regression by randomForest (2002). R News, 2(3):18-22. https://doi.org/10.1023/A:1010933404324

See Also

`randomForest`

cross.validation-class

S4 class to represent k-fold cross-validation results.

Description

S4 class to represent k-fold cross-validation results.

Slots

`DvsHC` Performances for Disease-Healthy classidication

`FTDvsNonFTD` Performances for FTD-NonFTD classidication

`ADvsDLB` Performances for AD-DLB classidication

performance

Evaluate classifier performances

Description

Compute performance indices for a binary classifier, from a vector of observed values and a vector of predictions.

Usage

```
performance(obs = NULL, pred = NULL, CT = NULL, y = "0,1")
```

Arguments

`obs`	Vector of observed values.
`pred`	Vector of predicted values.
`CT`	An object of class "table". Alternatively to the vectors of observed and predicted values, the user could provide a 2x2 contingency table.
`y`	Outcome encoding. Use "0,1" ("1,2") if the binary encoding is 0 (1) for controls and 1 (2) for cases.

Value

An object of class `performance`.

performance-class *S4 class to represent classifier performances.*

Description

S4 class to represent classifier performances.

Slots

`ctable` Confusion 2x2 table

`performance` List of classifier performance indices

rmout *Remove outliers using subject-wise Brier scores*

Description

Detect and remove outliers from a dataset using per subject Brier scores (Brier, 1950).

Usage

```
rmout(data, status = 10, k = 5, b = 1)
```

Arguments

`data`	A matrix or data.frame containing subjects as rows and TMS parameters as columns. A further column specifying subject diagnosis is required to run RFC cross-validation.
`status`	Numeric value specifying subject diagnosis (default status = 10).
`k`	Number of cross-validation cycles (default k = 5).
`b`	Maximum Brier score beyond which a subject is considered as an outlier. By default, b = 1. If b is set to NULL, outliers will be automatically set to Q3 + 1.5*(Q3 - Q1).

Value

The input data.frame without outliers.

References

Brier GW (1950). Verification of forecasts expressed in terms of probability. Monthly Weather Review, 78(1):1-3. https://doi.org/10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2

See Also

`evaluation`

tms.coef *Default TMS regression coefficients*

Description

Data.frame of TMS regression coefficients derived from default TMS data (Benussi, 2017). They include regression coefficients over time (t) for short-interval intracortical inhibition (SICI ~ bs0 + bs*t), intracortical facilitation (ICF ~ bi0 + bi*t), short-latency afferent inhibition (SAI ~ b0 + b1*t + b2*t^2), and long-interval intracortical inhibition (LICI ~ a0 + a1*t).

Usage

```
tms.coef
```

Format

"tms.coef" is a data.frame of 649 subjects (rows) and 9 TMS regression coefficients (columns). Subjects are subdivided into 273 Alzheimer's Disease patients (AD), 207 Frontotemporal Dementia patients (FTD), 67 Dementia with Lewy Bodies patients (DLB), and 147 healthy controls (HC).

References

Benussi A, Di Lorenzo F, Dell'Era V, Cosseddu M, Alberici A, Caratozzolo S, Cotelli MS, Micheli A, Rozzini L, Depari A, Flammini A, Ponzo V, Martorana A, Caltagirone C, Padovani A, Koch G, Borroni B. (2017). Transcranial magnetic stimulation distinguishes Alzheimer disease from frontotemporal dementia. Neurology, 89(7):665-672. https://doi.org/10.1212/WNL.0000000000004232

tms.rfc1 *D-HC Random Forest Classifier (RFC1)*

Description

This is the first of three RFCs used by tmsClssifier to predict subjects' disgnosis on the base of TMS data. The RFC1 classify subjects in either "Diseased" (AD, FTD, or DLB) or "Healthy" (i.e. the typical TMS signature of heathy controls). If a subject is classified as "Diseased", the classification process moves to the second classifier (RFC2). In case of an "Healthy" subject, the classification process terminates.

Usage

```
tms.rfc1
```

Format

"tms.rfc1" is an object of class `randomForest`.

tms.rfc2 *FTD-NonFTD Random Forest Classifier (RFC2)*

Description

This is the second of three RFCs used by tmsClssifier to predict subjects' disgnosis on the base of TMS data. For each RFC1 classification that yielded "Diseased", RFC2 decides if the respective subject is of "FTD" (Frontotemporal Dementia) or "Non-FTD" class. In case of an FTD subject, the classification process terminates.

Usage

```
tms.rfc2
```

Format

"tms.rfc2" is an object of class `randomForest`.

tms.rfc3 *AD-DLB Random Forest Classifier (RFC3)*

Description

This is the third of three RFCs used by tmsClssifier to predict subjects' disgnosis on the base of TMS data. For each RFC2 classification that yielded "Non-FTD", RFC3 decides if the respective subject is of "AD" (Alzheimer's Disease) or "DLB" (Dementia with Lewy Bodies) class.

Usage

```
tms.rfc3
```

Format

"tms.rfc3" is an object of class `randomForest`.

`tmsClassify` *TMS Random Forest Classifier (RFC)*

Description

Classify subjects with unknown diagnosis in one class among Healthy subject (HC), Frontotemporal Dementia (FTD), Alzheimer's Disease (AD), or Dementia with Lewy Bodies (DLB), using estimated TMS regression parameters. This function provides a default RFC, requiring an input data.frame of 9 columns (4 for SICI-ICF, 3 for SAI, and 2 for LICI; see `tmsRegression` for details). See function `buildPredictor` to build a custom RFC. Alternatively, if a diagnosis column is present in the input data.frame, it can be used to test RFC performances through cross-validation.

Usage

```
tmsClassify(data, cv = FALSE, status = 10, p = 1, k = 5)
```

Arguments

`data`	A data.frame containing subjects as rows and TMS parameters as columns. Optionally, a column specifying patient diagnosis can be added to test RFC performances using cross-validation.
`cv`	A logical value. If cv is TRUE, the cross-validation mode is enabled, and a status (i.e., diagnosis) column is needed (see below). By default, cv = FALSE and the diagnosis prediction mode is enabled.
`status`	Numeric value indicating the column for subject diagnosis (default status = 10). This value is ignored if cv is set to FALSE.
`p`	Proportion of the input dataset to be used for the cross-validation. For instance, if p = 0.75, a randomly chosen 25 the input data.frame will be isolated and returned as validation set, while the remaining 75 is balanced so that the original class proportion is preserved. By default, p = 1 (i.e., the entire dataset is used for cross-validation) and no validation set is generated.
`k`	Number of cross-validation iterations (default k = 5).

Value

If cv = FALSE, the vector of diagnosis prediction for each subject is returned. If cv = TRUE, an object of class cross.validation is generated, including cross-validation results. If p < 1, a list of three objects is returned:

1. "CV", cross-validation performances;
2. "trainingSet", dataset partition used for the cross-vaidation;
3. "validationSet", dataset partition isolated before cross-validation.

References

Slawski M, Daumer M, Boulesteix AL (2008). CMA - a comprehensive Bioconductor package for supervised classification with high dimensional data. BMC Bioinformatics, 9:439. https://doi.org/10.1186/1471-2105-9-439

See Also

GenerateLearningsets and classification.

`tmsImpute` *Missing data imputation*

Description

Impute missing Transcranial Magnetic Stimulation (TMS) values, using the K-nearest neighbours (KNN) algorithm from `impute.knn` (Troyanskaya, 2001).

Usage

```
tmsImpute(tms, sici.icf = 1:7, sai = 8:11, lici = 12:14,
  max.na = 7, max.r = 0.5, max.c = 0.8, k = 10)
```

Arguments

`tms`	A matrix or data.frame containing subjects as rows and TMS values as columns.
`sici.icf`	Numeric vector determining the position of temporally-ordered SICI-ICF columns (SICI: short-interval intracortical inhibition; ICF: intracortical facilitation). By default, they should be the first 7 measures (sici.icf = 1:7; 4 for SICI and 3 for ICF), taken at times (interstimulus intervals): 1, 2, 3, 5, 7, 10, 15 ms. Set sici.icf to NULL to exclude these values from classification.
`sai`	Numeric vector determining the position of temporally-ordered SAI (short-latency afferent inhibition) columns. By default, they should be the 4 columns following sici.icf (sai = 8:11), taken at time steps (interstimulus intervals): -4, 0, 4, 8 ms. Set sai to NULL to exclude these values from classification.
`lici`	Numeric vector determining the position of temporally-ordered LICI (long-interval intracortical inhibition) columns. By default, they should be the 3 columns following sai (lici = 12:14), taken at time steps (interstimulus intervals): 50, 100, 150 ms. Set lici to NULL to exclude these values from classification.
`max.na`	Maximum number of missing values per row (default max.na = 7).
`max.r`	Maximum percentage of missing values per row (default max.r = 0.5).
`max.c`	Maximum percentage of missing values per column (default max.c = 0.8).
`k`	Number of nearest neighbours to perform imputation.

Value

The original data.frame with imputed missing values.

References

Troyanskaya O, Cantor M, Sherlock G, Brown P, Hastie T, Tibshirani R, Botstein D, Altman RB (2001). Missing value estimation methods for DNA microarrays. Bioinformatics, 17(6):520-525. https://doi.org/10.1093/bioinformatics/17.6.520

See Also

`impute.knn` for further details on KNN imputation.

`tmsRegression` *Generate regression parameters from TMS data*

Description

Generate regression parameters from TMS data, needed for subject classification. For each subject, TMS indicators (SICI-ICF, SAI, LICI) are modeled as a polynomial functions of time, in the form y ~ poly(t). This function estimates two parameters for SICI (SICI = bs0 + bs*t), two parameters for ICF (ICF = bi0 + bi*t), three parameters for SAI (SAI = b0 + b1*t + b2*t^2), and two parameters for LICI (LICI = a0 + a1*t).

Usage

```
tmsRegression(tms, sici.icf = 1:7, sai = 8:11, lici = 12:14,
  adjust = NULL)
```

Arguments

`tms`	A data.frame containing subjects as rows and TMS values as columns. Optionally, the user may add covariate columns (e.g., sex, age, center) to adjust TMS regression estimates.
`sici.icf`	Numeric vector determining the position of temporally-ordered SICI-ICF columns (SICI: short-interval intracortical inhibition; ICF: intracortical facilitation). By default, they should be the first 7 measures (sici.icf = 1:7; 4 for SICI and 3 for ICF), taken at times (interstimulus intervals): 1, 2, 3, 5, 7, 10, 15 ms. Set sici.icf to NULL to exclude these values from classification.
`sai`	Numeric vector determining the position of temporally-ordered SAI (short-latency afferent inhibition) columns. By default, they should be the 4 columns following sici.icf (sai = 8:11), taken at time steps (interstimulus intervals): -4, 0, 4, 8 ms. Set sai to NULL to exclude these values from classification.
`lici`	Numeric vector determining the position of temporally-ordered LICI (long-interval intracortical inhibition) columns. By default, they should be the 3 columns following sai (lici = 12:14), taken at time steps (interstimulus intervals): 50, 100, 150 ms. Set lici to NULL to exclude these values from classification.
`adjust`	Numeric vector determining the position of covariates to adjust for. By default, adjust = NULL (no covariate adjustment is done).

Value

A data.frame of estimated regression parameters.

`vpart` *Data partitioning utility*

Description

Extract a random partition from an input dataset.

Usage

```
vpart(data, p, status = NULL, shuffle = FALSE)
```

Arguments

`data`	An input data.frame.
`p`	Proportion of input rows to be extracted from the input dataset.
`status`	Numeric value indicating the column for subject diagnosis (default status = NULL). Allowed values for this column are "HC", "FTD", "AD", and "DLB". If status is not NULL, random partitioning will preserve the status column proportions.
`shuffle`	A logical value. If TRUE, the input rows are randomly shuffled before data partitioning.

Value

A list of 2 data.frames:

1. "training.set", the portion of the input data defined by p;
2. "validation.set", the portion of the input data defined by 1-p.

Index