Package 'ECoL'

February 26, 2019

Type Package

Version 0.2.0

Date 2019-02-26

Title Complexity Measures for Supervised Problems

Description Provides measures to characterize the complexity of classification and regression problems based on aspects that quantify the linearity of the data, the presence of informative feature, the sparsity and dimensionality of the datasets. This package provides bug fixes, generalizations and implementations of many state of the art measures. The measures are described in the papers: Ho and Basu (2002) <doi:10.1109/34.990132> and Lorena et al. (2018) <doi:10.1007/s10994-017-5681-1>.

Author Luis Garcia [aut, cre],
Ana Lorena [aut, ctb]

Maintainer Luis Garcia <lpfgarcia@icmc.usp.br>

URL https://github.com/lpfgarcia/ECoL/

Imports cluster, e1071, FNN, igraph, MASS

License MIT + file LICENSE

BugReports https://github.com/lpfgarcia/ECoL/issues

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2019-02-26 21:10:16 UTC

R topics documented:

`balance` *Measures of class balance*

Description

Classification task. These measures capture the differences in the number of examples per class in the dataset. When these differences are severe, problems related to generalization of the ML classification techniques could happen because of the imbalance ratio.

Usage

```
balance(...)

## Default S3 method:
balance(x, y, measures = "all", ...)

## S3 method for class 'formula'
balance(formula, data, measures = "all", ...)
```

Arguments

`...`	Not used.
`x`	A data.frame contained only the input attributes.
`y`	A factor response vector with one label for each row/component of x.
`measures`	A list of measures names or `"all"` to include all them.
`formula`	A formula to define the class column.
`data`	A data.frame dataset contained the input attributes and class.

Details

The following measures are allowed for this method:

"C1" The entropy of class proportions (C1) capture the imbalance in a dataset based on the proportions of examples per class.

"C2" The imbalance ratio (C2) is an index computed for measuring class balance. This is a version of the measure that is also suited for multiclass classification problems.

Value

A list named by the requested class balance measure.

References

Ana C Lorena, Ivan G Costa, Newton Spolaor and Marcilio C P Souto. (2012). Analysis of complexity indices for classification problems: Cancer gene expression data. Neurocomputing 75, 1, 33–42.

Ajay K Tanwani and Muddassar Farooq. (2010). Classification potential vs. classification accuracy: a comprehensive study of evolutionary algorithms with biomedical datasets. Learning Classifier Systems 6471, 127–144.

See Also

Other complexity-measures: `correlation`, `dimensionality`, `linearity.class`, `linearity.regr`, `neighborhood`, `network`, `overlapping`, `smoothness`

Examples

```
## Extract all balance measures
data(iris)
balance(Species ~ ., iris)
```

`complexity` *Extract the complexity measures from datasets*

Description

This function is responsable to extract the complexity measures from the classification and regression tasks. For such, they take into account the overlap between classes imposed by feature values, the separability and distribution of the data points and the value of structural measures based on the representation of the dataset as a graph structure. To set specific parameters for each group, use the characterization function.

Usage

```
complexity(...)

## Default S3 method:
complexity(x, y, type, groups = "all", ...)

## S3 method for class 'formula'
complexity(formula, data, type, groups = "all", ...)
```

Arguments

...	Not used.
x	A data.frame contained only the input attributes.
y	A response vector with one value for each row/component of x.
type	The type of supervised problem: The "class" is used for classification tasks and "regr" for regression tasks.
groups	A list of complexity measures groups or "all" to include all of them.
formula	A formula to define the output column.
data	A data.frame dataset contained the input and output attributes.

Details

The following groups are allowed for this method:

"overlapping" The feature overlapping measures characterize how informative the available features are to separate the classes See overlapping for more details.

"neighborhood" Neighborhood measures characterize the presence and density of same or different classes in local neighborhoods. See neighborhood for more details.

"linearity" Linearity measures try to quantify whether the classes can be linearly separated. See linearity.class or linearity.regr for more details.

"dimensionality" The dimensionality measures compute information on how smoothly the examples are distributed within the attributes. See dimensionality for more details.

"balance" Class balance measures take into account the numbers of examples per class in the dataset. See balance for more details.

"network" Network measures represent the dataset as a graph and extract structural information from it. See network for more details.

"correlation" Capture the relationship of the feature values with the outputs. See correlation for more details.

"smoothness" Estimate the smoothness of the function that must be fitted to the data. See smoothness for more details.

Value

A numeric vector named by the requested complexity measures.

References

Tin K Ho and Mitra Basu. (2002). Complexity measures of supervised classification problems. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24, 3, 289–300.

Albert Orriols-Puig, Nuria Macia and Tin K Ho. (2010). Documentation for the data complexity library in C++. Technical Report. La Salle - Universitat Ramon Llull.

Ana C Lorena and Aron I Maciel and Pericles B C Miranda and Ivan G Costa and Ricardo B C Prudencio. (2018). Data complexity meta-features for regression problems. Machine Learning, 107, 1, 209–246.

Examples

```
## Extract all complexity measures for classification task
data(iris)
complexity(Species ~ ., iris, type="class")

## Extract all complexity measures for regression task
data(cars)
complexity(speed~., cars, type="regr")
```

`correlation` *Measures of feature correlation*

Description

Regression task. These measures calculate the correlation of the values of the features to the outputs. If at least one feature is highly correlated to the output, this indicates that simpler functions can be fitted to the data.

Usage

```
correlation(...)

## Default S3 method:
correlation(x, y, measures = "all", ...)

## S3 method for class 'formula'
correlation(formula, data, measures = "all", ...)
```

Arguments

`...`	Not used.
`x`	A data.frame contained only the input attributes.
`y`	A response vector with one value for each row/component of x.
`measures`	A list of measures names or `"all"` to include all them.
`formula`	A formula to define the output column.
`data`	A data.frame dataset contained the input and output attributes.

Details

The following measures are allowed for this method:

"C1" Maximum feature correlation to the output (C1) calculate the maximum absolute value of the Spearman correlation between each feature and the outputs.

"C2" Average feature correlation to the output (C2) computes the average of the Spearman correlations of all features to the output.

"C3" Individual feature efficiency (C3) calculates, for each feature, the number of examples that must be removed from the dataset until a high Spearman correlation value to the output is achieved.

"C4" Collective feature efficiency (C4) computes the ratio of examples removed from the dataset based on an iterative process of linear fitting between the features and the target attribute.

Value

A list named by the requested correlation measure.

References

Ana C Lorena and Aron I Maciel and Pericles B C Miranda and Ivan G Costa and Ricardo B C Prudencio. (2018). Data complexity meta-features for regression problems. Machine Learning, 107, 1, 209–246.

See Also

Other complexity-measures: balance, dimensionality, linearity.class, linearity.regr, neighborhood, network, overlapping, smoothness

Examples

```
## Extract all correlation measures
data(cars)
correlation(speed~., cars)
```

`dimensionality` *Measures of dimensionality*

Description

These measures give an indicative of data sparsity. They capture how sparse a datasets tend to have regions of low density. These regions are know to be more difficult to extract good classification and regression models.

Usage

```
dimensionality(...)

## Default S3 method:
dimensionality(x, y, measures = "all", ...)

## S3 method for class 'formula'
dimensionality(formula, data, measures = "all", ...)
```

Arguments

...	Not used.
x	A data.frame contained only the input attributes.
y	A response vector with one value for each row/component of x.
measures	A list of measures names or "all" to include all them.
formula	A formula to define the output column.
data	A data.frame dataset contained the input and output attributes.

Details

The following measures are allowed for this method:

"T2" Average number of points per dimension (T2) is given by the ratio between the number of examples and dimensionality of the dataset.

"T3" Average number of points per PCA (T3) is similar to T2, but uses the number of PCA components needed to represent 95 variability as the base of data sparsity assessment.

"T4" Ratio of the PCA Dimension to the Original (T4) estimates the proportion of relevant and the original dimensions for a dataset.

Value

A list named by the requested dimensionality measure.

References

Ana C Lorena, Ivan G Costa, Newton Spolaor and Marcilio C P Souto. (2012). Analysis of complexity indices for classification problems: Cancer gene expression data. Neurocomputing 75, 1, 33–42.

See Also

Other complexity-measures: balance, correlation, linearity.class, linearity.regr, neighborhood, network, overlapping, smoothness

Examples

```
## Extract all dimensionality measures
data(iris)
dimensionality(Species ~ ., iris)

data(cars)
correlation(speed~., cars)
```

`linearity.class` *Measures of linearity*

Description

Classification task. The linearity measures try to quantify if it is possible to separate the classes by a hyperplane. The underlying assumption is that a linearly separable problem can be considered simpler than a problem requiring a non-linear decision boundary.

Usage

```
linearity.class(...)

## Default S3 method:
linearity.class(x, y, measures = "all", ...)

## S3 method for class 'formula'
linearity.class(formula, data, measures = "all", ...)
```

Arguments

`...`	Not used.
`x`	A data.frame contained only the input attributes.
`y`	A factor response vector with one label for each row/component of `x`.
`measures`	A list of measures names or `"all"` to include all them.
`formula`	A formula to define the class column.
`data`	A data.frame dataset contained the input attributes and class.

Details

The following measures are allowed for this method:

"L1" Sum of the error distance by linear programming (L1) computes the sum of the distances of incorrectly classified examples to a linear boundary used in their classification.

"L2" Error rate of linear classifier (L2) computes the error rate of the linear SVM classifier induced from dataset.

"L3" Non-linearity of a linear classifier (L3) creates a new dataset randomly interpolating pairs of training examples of the same class and then induce a linear SVM on the original data and measure the error rate in the new data points.

Value

A list named by the requested linearity measure.

References

Albert Orriols-Puig, Nuria Macia and Tin K Ho. (2010). Documentation for the data complexity library in C++. Technical Report. La Salle - Universitat Ramon Llull.

See Also

Other complexity-measures: balance, correlation, dimensionality, linearity.regr, neighborhood, network, overlapping, smoothness

Examples

```
## Extract all linearity measures
data(iris)
linearity.class(Species ~ ., iris)
```

---

linearity.regr *Measures of linearity*

Description

Regression task. These measures capture whether a linear function provides a good fit to the problem. If this is the case, the problem can be considered simpler than one in which a non-linear function is required.

Usage

```
linearity.regr(...)

## Default S3 method:
linearity.regr(x, y, measures = "all", ...)

## S3 method for class 'formula'
linearity.regr(formula, data, measures = "all", ...)
```

Arguments

...	Not used.
x	A data.frame contained only the input attributes.
y	A response vector with one value for each row/component of x.
measures	A list of measures names or "all" to include all them.
formula	A formula to define the output column.
data	A data.frame dataset contained the input and output attributes.

Details

The following measures are allowed for this method:

"L1" Mean absolute error (L1) averages the absolute values of the residues of a multiple linear regressor.

"L2" Residuals variance (L2) averages the square of the residuals from a multiple linear regression.

"L3" Non-linearity of a linear regressor (L3) measures how sensitive the regressor is to the new randomly interpolated points.

Value

A list named by the requested linearity measure.

References

Ana C Lorena and Aron I Maciel and Pericles B C Miranda and Ivan G Costa and Ricardo B C Prudencio. (2018). Data complexity meta-features for regression problems. Machine Learning, 107, 1, 209–246.

See Also

Other complexity-measures: balance, correlation, dimensionality, linearity.class, neighborhood, network, overlapping, smoothness

Examples

```
## Extract all regression linearity measures
data(cars)
linearity.regr(speed~., cars)
```

`neighborhood` *Measures of neighborhood*

Description

Classification task. The Neighborhood measures analyze the neighborhoods of the data items and try to capture class overlapping and the shape of the decision boundary. They work over a distance matrix storing the distances between all pairs of data points in the dataset.

Usage

```
neighborhood(...)

## Default S3 method:
neighborhood(x, y, measures = "all", ...)

## S3 method for class 'formula'
neighborhood(formula, data, measures = "all", ...)
```

Arguments

...	Not used.
x	A data.frame contained only the input attributes.
y	A factor response vector with one label for each row/component of x.
measures	A list of measures names or "all" to include all them.
formula	A formula to define the class column.
data	A data.frame dataset contained the input attributes and class.

Details

The following measures are allowed for this method:

"N1" Fraction of borderline points (N1) computes the percentage of vertexes incident to edges connecting examples of opposite classes in a Minimum Spanning Tree (MST).

"N2" Ratio of intra/extra class nearest neighbor distance (N2) computes the ratio of two sums: intra-class and inter-class. The former corresponds to the sum of the distances between each example and its closest neighbor from the same class. The later is the sum of the distances between each example and its closest neighbor from another class (nearest enemy).

"N3" Error rate of the nearest neighbor (N3) classifier corresponds to the error rate of a one Nearest Neighbor (1NN) classifier, estimated using a leave-one-out procedure in dataset.

"N4" Non-linearity of the nearest neighbor classifier (N4) creates a new dataset randomly interpolating pairs of training examples of the same class and then induce a the 1NN classifier on the original data and measure the error rate in the new data points.

"T1" Fraction of hyperspheres covering data (T1) builds hyperspheres centered at each one of the training examples, which have their radios growth until the hypersphere reaches an example of another class. Afterwards, smaller hyperspheres contained in larger hyperspheres are eliminated. T1 is finally defined as the ratio between the number of the remaining hyperspheres and the total number of examples in the dataset.

"LSC" Local Set Average Cardinality (LSC) is based on Local Set (LS) and defined as the set of points from the dataset whose distance of each example is smaller than the distance from the exemples of the different class. LSC is the average of the LS.

Value

A list named by the requested neighborhood measure.

References

Albert Orriols-Puig, Nuria Macia and Tin K Ho. (2010). Documentation for the data complexity library in C++. Technical Report. La Salle - Universitat Ramon Llull.

Enrique Leyva, Antonio Gonzalez and Raul Perez. (2014). A Set of Complexity Measures Designed for Applying Meta-Learning to Instance Selection. IEEE Transactions on Knowledge and Data Engineering 27, 2, 354–367.

See Also

Other complexity-measures: balance, correlation, dimensionality, linearity.class, linearity.regr, network, overlapping, smoothness

Examples

```
## Extract all neighborhood measures
data(iris)
neighborhood(Species ~ ., iris)
```

`network` *Measures of network*

Description

Classification task. The network measures represent the dataset as a graph and extract structural information from it. The transformation between raw data and the graph representation is based on the epsilon-NN algorithm. Next, a post-processing step is applied to the graph, pruning edges between examples of opposite classes.

Usage

```
network(...)

## Default S3 method:
network(x, y, measures = "all", eps = 0.15, ...)

## S3 method for class 'formula'
network(formula, data, measures = "all", eps = 0.15,
  ...)
```

Arguments

`...`	Not used.
`x`	A data.frame contained only the input attributes.
`y`	A factor response vector with one label for each row/component of `x`.
`measures`	A list of measures names or `"all"` to include all them.
`eps`	The percentage of nodes in the graph to be connected.
`formula`	A formula to define the class column.
`data`	A data.frame dataset contained the input attributes and class.

Details

The following measures are allowed for this method:

"Density" Average Density of the network (Density) represents the number of edges in the graph, divided by the maximum number of edges between pairs of data points.

"ClsCoef" Clustering coefficient (ClsCoef) averages the clustering tendency of the vertexes by the ratio of existent edges between its neighbors and the total number of edges that could possibly exist between them.

"Hubs" Hubs score (Hubs) is given by the number of connections it has to other nodes, weighted by the number of connections these neighbors have.

Value

A list named by the requested network measure.

References

Gleison Morais and Ronaldo C Prati. (2013). Complex Network Measures for Data Set Characterization. In 2nd Brazilian Conference on Intelligent Systems (BRACIS). 12–18.

Luis P F Garcia, Andre C P L F de Carvalho and Ana C Lorena. (2015). Effect of label noise in the complexity of classification problems. Neurocomputing 160, 108–119.

See Also

Other complexity-measures: `balance`, `correlation`, `dimensionality`, `linearity.class`, `linearity.regr`, `neighborhood`, `overlapping`, `smoothness`

Examples

```
## Extract all network measures
data(iris)
network(Species ~ ., iris)
```

`overlapping` *Measures of overlapping*

Description

Classification task. The overlapping measures evaluate how informative the available features are to separate the classes. If there is at least one very discriminative feature in the dataset, the problem can be considered simpler than if there is no such an attribute.

Usage

```
overlapping(...)

## Default S3 method:
overlapping(x, y, measures = "all", ...)

## S3 method for class 'formula'
overlapping(formula, data, measures = "all", ...)
```

Arguments

`...`	Not used.
`x`	A data.frame contained only the input attributes.
`y`	A factor response vector with one label for each row/component of x.
`measures`	A list of measures names or `"all"` to include all them.
`formula`	A formula to define the class column.
`data`	A data.frame dataset contained the input attributes and class.

Details

The following measures are allowed for this method:

"F1" Maximum Fisher's Discriminant Ratio (F1) measures the overlap between the values of the features and takes the value of the largest discriminant ratio among all the available features.

"F1v" Directional-vector maximum Fisher's discriminant ratio (F1v) complements F1 by searching for a vector able to separate two classes after the training examples have been projected into it.

"F2" Volume of the overlapping region (F2) computes the overlap of the distributions of the features values within the classes. F2 can be determined by finding, for each feature its minimum and maximum values in the classes.

"F3" The maximum individual feature efficiency (F3) of each feature is given by the ratio between the number of examples that are not in the overlapping region of two classes and the total number of examples. This measure returns the maximum of the values found among the input features.

"F4" Collective feature efficiency (F4) get an overview on how various features may work together in data separation. First the most discriminative feature according to F3 is selected and all examples that can be separated by this feature are removed from the dataset. The previous step is repeated on the remaining dataset until all the features have been considered or no example remains. F4 returns the ratio of examples that have been discriminated.

Value

A list named by the requested overlapping measure.

References

Albert Orriols-Puig, Nuria Macia and Tin K Ho. (2010). Documentation for the data complexity library in C++. Technical Report. La Salle - Universitat Ramon Llull.

See Also

Other complexity-measures: balance, correlation, dimensionality, linearity.class, linearity.regr, neighborhood, network, smoothness

Examples

```
## Extract all overlapping measures
data(iris)
overlapping(Species ~ ., iris)
```

`smoothness` *Measures of smoothness*

Description

Regression task. In regression problems, the smoother the function to be fitted to the data, the simpler it shall be. Larger variations in the inputs and/or outputs, on the other hand, usually indicate the existence of more intricate relationships between them.

Usage

```
smoothness(...)

## Default S3 method:
smoothness(x, y, measures = "all", ...)

## S3 method for class 'formula'
smoothness(formula, data, measures = "all", ...)
```

Arguments

`...`	Not used.
`x`	A data.frame contained only the input attributes.
`y`	A response vector with one value for each row/component of x.
`measures`	A list of measures names or `"all"` to include all them.
`formula`	A formula to define the output column.
`data`	A data.frame dataset contained the input and output attributes.

Details

The following measures are allowed for this method:

"S1" Output distribution (S1) monitors whether the examples joined in the MST have similar output values. Lower values indicate simpler problems, where the outputs of similar examples in the input space are also next to each other.

"S2" Input distribution (S2) measure how similar in the input space are data items with similar outputs based on distance.

"S3" Error of a nearest neighbor regressor (S3) calculates the mean squared error of a 1-nearest neighbor regressor using leave-one-out.

"S4" Non-linearity of nearest neighbor regressor (S4) calculates the mean squared error of a 1-nearest neighbor regressor to the new randomly interpolated points.

Value

A list named by the requested smoothness measure.

References

Ana C Lorena and Aron I Maciel and Pericles B C Miranda and Ivan G Costa and Ricardo B C Prudencio. (2018). Data complexity meta-features for regression problems. Machine Learning, 107, 1, 209–246.

See Also

Other complexity-measures: `balance`, `correlation`, `dimensionality`, `linearity.class`, `linearity.regr`, `neighborhood`, `network`, `overlapping`

Examples

```
## Extract all smoothness measures
data(cars)
smoothness(speed~., cars)
```

Index