Package ‘scutr’

February 8, 2021

Title Balancing Multiclass Datasets for Classification Tasks

Version 0.1

Description Imbalanced training datasets impede many popular classifiers. To balance training data, a combination of oversampling minority classes and undersampling majority classes is necessary. This package implements the SCUT (SMOTE and Cluster-based Undersampling Technique) algorithm, which uses model-based clustering and synthetic oversampling to balance multiclass training datasets.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Imports smotefamily,
doParallel,
parallel,
foreach,
mclust

Depends R (>= 2.10)

URL https://github.com/s-kganz/scutr

BugReports https://github.com/s-kganz/scutr/issues

Suggests testthat (>= 2.0.0)

Config/testthat/edition 2

R topics documented:

bullseye 2
imbalance 2
oversample.smote 3
resample.random 3
sample.classes 4
SCUT 4
undersample.hclust 6
undersample.kmeans 7
undersample.mclust 7
undersample.mindist 8

undersample.tomek 9
validate.dataset 10
wine 10

Index **11**

`bullseye` *An imbalanced dataset with a minor class centered around the origin with a majority class surrounding the center.*

Description

An imbalanced dataset with a minor class centered around the origin with a majority class surrounding the center.

Usage

```
bullseye
```

Format

a data.frame with 1000 rows and 3 columns.

Source

https://gist.github.com/s-kganz/c2534666e369f8e19491bb29d53c619d

---

`imbalance` *An imbalanced dataset with randomly placed normal distributions around the origin. The nth class has n * 10 observations.*

Description

An imbalanced dataset with randomly placed normal distributions around the origin. The nth class has n * 10 observations.

Usage

```
imbalance
```

Format

a data.frame with 2100 rows and 11 columns

Source

https://gist.github.com/s-kganz/d08473f9492d48ea0e56c3c8a3fe1a74

`oversample.smote` *Oversample a dataset by SMOTE.*

Description

Oversample a dataset by SMOTE.

Usage

```
oversample.smote(data, cls, cls.col, m)
```

Arguments

`data`	Dataset to be oversampled.
`cls`	Class to be oversampled.
`cls.col`	Column containing class information.
`m`	Desired number of samples in the oversampled data.

Value

The oversampled dataset.

Examples

```
table(iris$Species)
smoted <- oversample.smote(iris, "setosa", "Species", 100)
nrow(smoted)
```

`resample.random` *Randomly resample a dataset.*

Description

This function is used to resample a dataset by randomly removing or duplicating rows. It is usable for both oversampling and undersampling.

Usage

```
resample.random(data, cls, cls.col, m)
```

Arguments

`data`	Dataframe to be resampled.
`cls`	Class that should be randomly resampled.
`cls.col`	Column containing class information.
`m`	Desired number of samples.

Value

Resampled dataframe containing only `cls`.

Examples

```
set.seed(1234)
only2 <- resample.random(wine, 2, "type", 15)
```

`sample.classes` *Stratified index sample of different values in a vector.*

Description

Stratified index sample of different values in a vector.

Usage

```
sample.classes(vec, tot.sample)
```

Arguments

`vec`	Vector of values to sample from.
`tot.sample`	Total number of samples.

Value

A vector of indices that can be used to select a balanced population of values from `vec`.

Examples

```
vec <- sample(1:5, 30, replace=TRUE)
table(vec)
sample.ind <- sample.classes(vec, 15)
table(vec[sample.ind])
```

`SCUT` *SMOTE and cluster-based undersampling technique.*

Description

This function balances multiclass training datasets. In a dataframe with `n` classes and `m` rows, the resulting dataframe will have `m / n` rows per class. `SCUT.parallel()` distributes each over/undersampling task across multiple cores. Speedup usually occurs only if there are many classes using one of the slower resampling techniques (e.g. `mclust`).

Usage

```
SCUT(
  data,
  cls.col,
  oversample = oversample.smote,
  undersample = undersample.mclust,
  osamp.opts = list(),
  usamp.opts = list()
)

SCUT.parallel(
  data,
  cls.col,
  ncores = detectCores()%/%2,
  oversample = oversample.smote,
  undersample = undersample.mclust,
  osamp.opts = list(),
  usamp.opts = list()
)
```

Arguments

`data` Numeric data frame containing all variables given in `form`.

`cls.col` The column in `data` with class membership.

`oversample` Oversampling method. Must be a function with the signature foo(data, cls, cls.col, m, ...
that returns a data frame, one of the oversample.* functions, or `sample.random`.

`undersample` Undersampling method. Must be a function with the signature `foo(data,cls,cls.col,m,...)`
that returns a data frame, one of the undersample.* functions, or `sample.random`.

`osamp.opts` Custom options passed to the oversampling function.

`usamp.opts` Custom options passed to the undersampling function.

`ncores` Number of cores to use with `SCUT.parallel`.

Value

A dataframe with equal class distribution.

Examples

```
ret <- SCUT(iris, "Species")
ret2 <- SCUT(chickwts, "feed", undersample=undersample.kmeans)
table(ret$Species)
table(ret2$feed)
ret <- SCUT.parallel(wine, "type", ncores=2, undersample=undersample.kmeans)
table(ret$type)
```

undersample.hclust *Undersample a dataset by hierarchical clustering.*

Description

Undersample a dataset by hierarchical clustering.

Usage

```
undersample.hclust(
  data,
  cls,
  cls.col,
  m,
  k = 5,
  h = NA,
  dist.calc = "euclidean"
)
```

Arguments

data	Dataset to be undersampled.
cls	Majority class that will be undersampled.
cls.col	Column in data containing class memberships.
m	Number of samples in undersampled dataset.
k	Number of clusters to derive from clustering.
h	Height at which to cut the clustering tree. k must be NA for this to be used.
dist.calc	Distance calculation method. See dist.

Value

Undersampled dataframe containing only cls.

Examples

```
table(iris$Species)
undersamp <- undersample.hclust(iris, "setosa", "Species", 15)
nrow(undersamp)
```

`undersample.kmeans` *Undersample a dataset by kmeans clustering.*

Description

Undersample a dataset by kmeans clustering.

Usage

```
undersample.kmeans(data, cls, cls.col, m, k = 5)
```

Arguments

`data`	Dataset to be undersampled.
`cls`	Class to be undersampled.
`cls.col`	Column containing class information.
`m`	Number of samples in undersampled dataset.
`k`	Number of centers in clustering.

Value

The undersampled dataframe containing only instances of `cls`.

Examples

```
table(iris$Species)
undersamp <- undersample.kmeans(iris, "setosa", "Species", 15)
nrow(undersamp)
```

`undersample.mclust` *Undersample a dataset by expectation-maximization clustering*

Description

Undersample a dataset by expectation-maximization clustering

Usage

```
undersample.mclust(data, cls, cls.col, m)
```

Arguments

`data`	Data to be undersampled.
`cls`	Class to be undersampled.
`cls.col`	Class column.
`m`	Number of samples in undersampled dataset.

Value

The undersampled dataframe containing only instance of `cls`.

Examples

```
setosa <- iris[iris$Species == "setosa", ]
nrow(setosa)
undersamp <- undersample.mclust(setosa, "setosa", "Species", 15)
nrow(undersamp)
```

`undersample.mindist` *Undersample a dataset by iteratively removing the observation with the lowest total distance to its neighbors of the same class.*

Description

Undersample a dataset by iteratively removing the observation with the lowest total distance to its neighbors of the same class.

Usage

```
undersample.mindist(data, cls, cls.col, m, dist.calc = "euclidean")
```

Arguments

`data`	Dataset to undersample. Aside from `cls.col`, must be numeric.
`cls`	Unused, but kept here for compatability with `do.undersample()`.
`cls.col`	Column containing class information.
`m`	Desired number of observations after undersampling.
`dist.calc`	Method for distance calculation. See `dist()`.

Value

An undersampled dataframe.

Examples

```
setosa <- iris[iris$Species == "setosa", ]
nrow(setosa)
undersamp <- undersample.mindist(setosa, "setosa", "Species", 50)
nrow(undersamp)
```

undersample.tomek *Undersample a dataset by removing Tomek links.*

Description

A Tomek link is a minority instance and majority instance that are each other's nearest neighbor. This function removes sufficient Tomek links that are an instance of cls to yield m instances of cls. If desired, samples are randomly discarded to yield m rows if insufficient Tomek links are in the data.

Usage

```
undersample.tomek(
  data,
  cls,
  cls.col,
  m,
  tomek = "minor",
  force.m = T,
  dist.calc = "euclidean"
)
```

Arguments

`data` Dataset to be undersampled.

`cls` Majority class to be undersampled.

`cls.col` Column in data containing class memberships.

`m` Desired number of samples in undersampled dataset.

`tomek` Definition used to determine if a point is considered a minority in the Tomek link definition.

- `minor`: Minor classes are all those with fewer than `m` instances.
- `diff`: Minor classes are all those that aren't `cls`.

`force.m` If `TRUE`, uses random undersampling to discard samples if insufficient Tomek links are present to yield `m` rows of data.

`dist.calc` Distance calculation method. See `dist`.

Value

Undersampled dataframe containing only `cls`.

Examples

```
table(iris$Species)
undersamp <- undersample.tomek(iris, "setosa", "Species", 15, tomek="diff", force.m=TRUE)
nrow(undersamp)
undersamp2 <- undersample.tomek(iris, "setosa", "Species", 15, tomek="diff", force.m=FALSE)
nrow(undersamp2)
```

`validate.dataset` *Validate a dataset for resampling.*

Description

This functions checks that the given column is present in the data and that all columns besides the class column are numeric.

Usage

```
validate.dataset(data, cls.col)
```

Arguments

`data`	Dataframe to validate.
`cls.col`	Column with class information.

Value

NA

`wine` *Type and chemical analysis of three different kinds of wine.*

Description

Type and chemical analysis of three different kinds of wine.

Usage

```
wine
```

Format

a data.frame with 178 rows and 14 columns

Source

https://archive.ics.uci.edu/ml/datasets/Wine

Index

*Topic **datasets**

- bullseye, 2
- imbalance, 2
- wine, 10

bullseye, 2

imbalance, 2

oversample.smote, 3

resample.random, 3

sample.classes, 4
SCUT, 4

undersample.hclust, 6
undersample.kmeans, 7
undersample.mclust, 7
undersample.mindist, 8
undersample.tomek, 9

validate.dataset, 10

wine, 10