Package: SMOTEWB (via r-universe)

August 23, 2024

Type Package

Title Imbalanced Resampling using SMOTE with Boosting (SMOTEWB)

Version 1.2.0

Description Provides the SMOTE with Boosting (SMOTEWB) algorithm. See F. Sağlam, M. A. Cengiz (2022) <doi:10.1016/j.eswa.2022.117023>. It is a SMOTE-based resampling technique which creates synthetic data on the links between nearest neighbors. SMOTEWB uses boosting weights to determine where to generate new samples and automatically decides the number of neighbors for each sample. It is robust to noise and outperforms most of the alternatives according to Matthew Correlation Coefficient metric. Alternative resampling methods are also available in the package.

Depends R (>= 4.2)

Imports stats, FNN, RANN, rpart, Rfast

License MIT + file LICENSE

Encoding UTF-8

LazyData false

RoxygenNote 7.3.1

Repository https://fatihsaglam.r-universe.dev

RemoteUrl https://github.com/fatihsaglam/smotewb

RemoteRef HEAD

RemoteSha ecf178dbf9b9fa785d9ce37c6bf23cc52aad3ee4

Contents

`ADASYN` *Adaptive Synthetic Sampling*

Description

Generates synthetic data for minority class to balance imbalanced datasets using ADASYN.

Usage

```
ADASYN(x, y, k = 5)
```

Arguments

`x`	feature matrix or data.frame.
`y`	a factor class variable with two classes.
`k`	number of neighbors. Default is 5.

Details

Adaptive Synthetic Sampling (ADASYN) is an extension of the Synthetic Minority Over-sampling Technique (SMOTE) algorithm, which is used to generate synthetic examples for the minority class (He et al., 2008). In contrast to SMOTE, ADASYN adaptively generates synthetic examples by focusing on the minority class examples that are harder to learn, meaning those examples that are closer to the decision boundary.

Note: Much faster than `smotefamily::ADAS()`.

Value

a list with resampled dataset.

`x_new`	Resampled feature matrix.
`y_new`	Resampled target variable.
`x_syn`	Generated synthetic data.
`C`	Number of synthetic samples for each positive class samples.

Author(s)

Fatih Saglam, saglamf89@gmail.com

References

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008, June). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence) (pp. 1322-1328). IEEE.

Examples

```
set.seed(1)
x <- rbind(matrix(rnorm(2000, 3, 1), ncol = 2, nrow = 1000),
           matrix(rnorm(100, 5, 1), ncol = 2, nrow = 50))
y <- as.factor(c(rep("negative", 1000), rep("positive", 50)))

plot(x, col = y)

# resampling
m <- ADASYN(x = x, y = y, k = 3)

plot(m$x_new, col = m$y_new)
```

BLSMOTE *Borderline Synthetic Minority Oversampling Technique*

Description

`BLSMOTE()` applies BLSMOTE (Borderline-SMOTE) which is a variation of the SMOTE algorithm that generates synthetic samples only in the vicinity of the borderline instances in imbalanced datasets.

Usage

```
BLSMOTE(x, y, k1 = 5, k2 = 5, type = "type1")
```

Arguments

`x`	feature matrix or data.frame.
`y`	a factor class variable with two classes.
`k1`	number of neighbors to link. Default is 5.
`k2`	number of neighbors to determine safe levels. Default is 5.
`type`	"type1" or "type2". Default is "type1".

Details

BLSMOTE works by focusing on the instances that are near the decision boundary between the minority and majority classes, known as borderline instances. These instances are more informative and potentially more challenging for classification, and thus generating synthetic samples in their vicinity can be more effective than generating them randomly.

Note: Much faster than `smotefamily::BLSMOTE()`.

Value

a list with resampled dataset.

x_new	Resampled feature matrix.
y_new	Resampled target variable.
x_syn	Generated synthetic data.
C	Number of synthetic samples for each positive class samples.

Author(s)

Fatih Saglam, saglamf89@gmail.com

References

Han, H., Wang, W. Y., & Mao, B. H. (2005). Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In Advances in Intelligent Computing: International Conference on Intelligent Computing, ICIC 2005, Hefei, China, August 23-26, 2005, Proceedings, Part I 1 (pp. 878-887). Springer Berlin Heidelberg.

Examples

```
set.seed(1)
x <- rbind(matrix(rnorm(2000, 3, 1), ncol = 2, nrow = 1000),
           matrix(rnorm(100, 5, 1), ncol = 2, nrow = 50))
y <- as.factor(c(rep("negative", 1000), rep("positive", 50)))

plot(x, col = y)

# resampling
m <- BLSMOTE(x = x, y = y, k1 = 5, k2 = 5)

plot(m$x_new, col = m$y_new)
```

GSMOTE *Geometric Synthetic Minority Oversamplnig Technique (GSMOTE)*

Description

Resampling with GSMOTE.

Usage

```
GSMOTE(x, y, k = 5, alpha_sel = "combined", alpha_trunc = 0.5, alpha_def = 0.5)
```

Arguments

x	feature matrix.
y	a factor class variable with two classes.
k	number of neighbors. Default is 5.
alpha_sel	selection method. Can be "minority", "majority" or "combined". Default is "combined".
alpha_trunc	truncation factor. A numeric value in $[-1, 1]$. Default is 0.5.
alpha_def	deformation factor. A numeric value in $[0, 1]$. Default is 0.5

Details

GSMOTE (Douzas & Bacao, 2019) is an oversampling method which creates synthetic samples geometrically around selected minority samples. Details are in the paper (Douzas & Bacao, 2019).

NOTE: Can not work with classes more than 2. Only numerical variables are allowed.

Value

a list with resampled dataset.

x_new	Resampled feature matrix.
y_new	Resampled target variable.
x_syn	Generated synthetic feature data.
y_syn	Generated synthetic label data.

Author(s)

Fatih Saglam, saglamf89@gmail.com

References

Douzas, G., & Bacao, F. (2019). Geometric SMOTE a geometrically enhanced drop-in replacement for SMOTE. Information sciences, 501, 118-135.

Examples

```
set.seed(1)
x <- rbind(matrix(rnorm(2000, 3, 1), ncol = 2, nrow = 1000),
           matrix(rnorm(100, 5, 1), ncol = 2, nrow = 50))
y <- as.factor(c(rep("negative", 1000), rep("positive", 50)))

plot(x, col = y)

# resampling
m <- GSMOTE(x = x, y = y, k = 7)

plot(m$x_new, col = m$y_new)
```

ROS *Random Oversampling (ROS)*

Description

Resampling with ROS.

Usage

```
ROS(x, y)
```

Arguments

x	feature matrix.
y	a factor class variable with two classes.

Details

Random Oversampling (ROS) is a method of copying and pasting of positive samples until balance is achieved.

Can work with classes more than 2.

Value

a list with resampled dataset.

x_new	Resampled feature matrix.
y_new	Resampled target variable.

Author(s)

Fatih Saglam, saglamf89@gmail.com

Examples

```
set.seed(1)
x <- rbind(matrix(rnorm(2000, 3, 1), ncol = 2, nrow = 1000),
           matrix(rnorm(100, 5, 1), ncol = 2, nrow = 50))
y <- as.factor(c(rep("negative", 1000), rep("positive", 50)))

plot(x, col = y)

# resampling
m <- ROS(x = x, y = y)

plot(m$x_new, col = m$y_new)
```

ROSE *Randomly Over Sampling Examples*

Description

Generates synthetic data for each class to balance imbalanced datasets using kernel density estimations. Can be used for multiclass datasets.

Usage

```
ROSE(x, y, h = 1)
```

Arguments

x	feature matrix or data.frame.
y	a factor class variable. Can be multiclass.
h	A numeric vector of length one or number of classes in y. If one is given, all classes will have same shrink factor. If a value is given for each classes, it will match respectively to `levels(y)`. Default is 1.

Details

Randomly Over Sampling Examples (ROSE) (Menardi and Torelli, 2014) is an oversampling method which uses conditional kernel densities to balance dataset. There is already an R package called 'ROSE' (Lunardon et al., 2014). Difference is that this one is much faster and can be applied for more than two classes.

Value

a list with resampled dataset.

x_new	Resampled feature matrix.
y_new	Resampled target variable.

Author(s)

Fatih Saglam, saglamf89@gmail.com

References

Lunardon, N., Menardi, G., and Torelli, N. (2014). ROSE: a Package for Binary Imbalanced Learning. R Jorunal, 6:82–92.

Menardi, G. and Torelli, N. (2014). Training and assessing classification rules with imbalanced data. Data Mining and Knowledge Discovery, 28:92–122.

Examples

```
set.seed(1)
x <- rbind(matrix(rnorm(2000, 3, 1), ncol = 2, nrow = 1000),
           matrix(rnorm(100, 5, 1), ncol = 2, nrow = 50))
y <- as.factor(c(rep("negative", 1000), rep("positive", 50)))

plot(x, col = y)

# resampling
m <- ROSE(x = x, y = y, h = c(0.12, 1))

plot(m$x_new, col = m$y_new)
```

RSLSMOTE *Relocating safe-level SMOTE with minority outcast handling*

Description

The Relocating Safe-Level SMOTE (RSLS) algorithm improves the quality of synthetic samples generated by Safe-Level SMOTE (SLS) by relocating specific synthetic data points that are too close to the majority class distribution towards the original minority class distribution in the feature space.

Usage

```
RSLSMOTE(x, y, k1 = 5, k2 = 5)
```

Arguments

x	feature matrix or data.frame.
y	a factor class variable with two classes.
k1	number of neighbors to link. Default is 5.
k2	number of neighbors to determine safe levels. Default is 5.

Details

In Safe-level SMOTE (SLS), a safe-level threshold is used to control the number of synthetic samples generated from each minority instance. This threshold is calculated based on the number of minority and majority instances in the local neighborhood of each minority instance. SLS generates synthetic samples that are located closer to the original minority class distribution in the feature space.

In Relocating safe-level SMOTE (RSLS), after generating synthetic samples using the SLS algorithm, the algorithm relocates specific synthetic data points that are deemed to be too close to the majority class distribution in the feature space. The relocation process moves these synthetic data points towards the original minority class distribution in the feature space.

This relocation process is performed by first identifying the synthetic data points that are too close to the majority class distribution. Then, for each identified synthetic data point, the algorithm calculates a relocation vector based on the distance between the synthetic data point and its k nearest minority class instances. This relocation vector is used to move the synthetic data point towards the minority class distribution in the feature space.

Note: Much faster than `smotefamily::RSLS()`.

Value

a list with resampled dataset.

`x_new`	Resampled feature matrix.
`y_new`	Resampled target variable.
`x_syn`	Generated synthetic data.
`C`	Number of synthetic samples for each positive class samples.

Author(s)

Fatih Saglam, saglamf89@gmail.com

References

Siriseriwan, W., & Sinapiromsaran, K. (2016). The effective redistribution for imbalance dataset: Relocating safe-level SMOTE with minority outcast handling. Chiang Mai J. Sci, 43(1), 234-246.

Examples

```
set.seed(1)
x <- rbind(matrix(rnorm(2000, 3, 1), ncol = 2, nrow = 1000),
           matrix(rnorm(100, 5, 1), ncol = 2, nrow = 50))
y <- as.factor(c(rep("negative", 1000), rep("positive", 50)))

plot(x, col = y)

# resampling
m <- RSLSMOTE(x = x, y = y, k1 = 5, k2 = 5)

plot(m$x_new, col = m$y_new)
```

`RUS` *Random Undersampling (RUS)*

Description

Resampling with RUS.

Usage

```
RUS(x, y)
```

Arguments

x	feature matrix.
y	a factor class variable with two classes.

Details

Random Undersampling (RUS) is a method of removing negative samples until balance is achieved.

Can work with classes more than 2.

Value

a list with resampled dataset.

x_new	Resampled feature matrix.
y_new	Resampled target variable.

Author(s)

Fatih Saglam, saglamf89@gmail.com

Examples

```
set.seed(1)
x <- rbind(matrix(rnorm(2000, 3, 1), ncol = 2, nrow = 1000),
           matrix(rnorm(100, 5, 1), ncol = 2, nrow = 50))
y <- as.factor(c(rep("negative", 1000), rep("positive", 50)))

plot(x, col = y)

# resampling
m <- RUS(x = x, y = y)

plot(m$x_new, col = m$y_new)
```

RWO *Random Walk Oversampling (SMOTE)*

Description

Resampling with RWO

Usage

```
RWO(x, y)
```

Arguments

x	feature matrix.
y	a factor class variable with two classes.

Details

RWO (Zhang and Li, 2014) is an oversampling method which generates data using variable standard error in a way that it preserves the variances of all variables.

Can work with classes more than 2.

Value

a list with resampled dataset.

x_new	Resampled feature matrix.
y_new	Resampled target variable.
x_syn	Generated synthetic feature data.
y_syn	Generated synthetic label data.

Author(s)

Fatih Saglam, saglamf89@gmail.com

References

Zhang, H., & Li, M. (2014). RWO-Sampling: A random walk over-sampling approach to imbalanced data classification. Information Fusion, 20, 99-116.

Examples

```
set.seed(1)
x <- rbind(matrix(rnorm(2000, 3, 1), ncol = 2, nrow = 1000),
           matrix(rnorm(100, 5, 1), ncol = 2, nrow = 50))
y <- as.factor(c(rep("negative", 1000), rep("positive", 50)))

plot(x, col = y)

# resampling
m <- RWO(x = x, y = y)

plot(m$x_new, col = m$y_new)
```

SLSMOTE *Safe-level Synthetic Minority Oversampling Technique*

Description

SLSMOTE() generates synthetic samples by considering a safe level of the nearest minority class examples.

Usage

```
SLSMOTE(x, y, k1 = 5, k2 = 5)
```

Arguments

x	feature matrix or data.frame.
y	a factor class variable with two classes.
k1	number of neighbors to link. Default is 5.
k2	number of neighbors to determine safe levels. Default is 5.

Details

SLSMOTE uses the safe-level distance metric to identify the minority class samples that are safe to oversample. Safe-level distance measures the distance between a minority class sample and its k-nearest minority class neighbors. A sample is considered safe to oversample if its safe-level is greater than a threshold. The safe-level of a sample is the ratio of minority class samples among its k-nearest neighbors.

In SLSMOTE, the oversampling process only applies to the safe minority class samples, which avoids the generation of noisy samples that can lead to overfitting. To generate synthetic samples, SLSMOTE randomly selects a minority class sample and finds its k-nearest minority class neighbors. Then, a random minority class neighbor is selected, and a synthetic sample is generated by adding a random proportion of the difference between the selected sample and its neighbor to the selected sample.

Note: Much faster than smotefamily::SLS().

Value

a list with resampled dataset.

x_new	Resampled feature matrix.
y_new	Resampled target variable.
x_syn	Generated synthetic data.
C	Number of synthetic samples for each positive class samples.

Author(s)

Fatih Saglam, saglamf89@gmail.com

References

Bunkhumpornpat, C., Sinapiromsaran, K., & Lursinsap, C. (2009). Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In Advances in Knowledge Discovery and Data Mining: 13th Pacific-Asia Conference, PAKDD 2009 Bangkok, Thailand, April 27-30, 2009 Proceedings 13 (pp. 475-482). Springer Berlin Heidelberg.

Examples

```
set.seed(1)
x <- rbind(matrix(rnorm(2000, 3, 1), ncol = 2, nrow = 1000),
           matrix(rnorm(100, 5, 1), ncol = 2, nrow = 50))
y <- as.factor(c(rep("negative", 1000), rep("positive", 50)))

plot(x, col = y)

# resampling
m <- SLSMOTE(x = x, y = y, k1 = 5, k2 = 5)

plot(m$x_new, col = m$y_new)
```

SMOTE *Synthetic Minority Oversampling Technique (SMOTE)*

Description

Resampling with SMOTE.

Usage

```
SMOTE(x, y, k = 5)
```

Arguments

x	feature matrix.
y	a factor class variable with two classes.
k	number of neighbors. Default is 5.

Details

SMOTE (Chawla et al., 2002) is an oversampling method which creates links between positive samples and nearest neighbors and generates synthetic samples along that link.

It is well known that SMOTE is sensitive to noisy data. It may create more noise.

Can work with classes more than 2.

Note: Much faster than `smotefamily::SMOTE()`.

Value

a list with resampled dataset.

x_new	Resampled feature matrix.
y_new	Resampled target variable.
x_syn	Generated synthetic feature data.
y_syn	Generated synthetic label data.

Author(s)

Fatih Saglam, saglamf89@gmail.com

References

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.

Examples

```
set.seed(1)
x <- rbind(matrix(rnorm(2000, 3, 1), ncol = 2, nrow = 1000),
           matrix(rnorm(100, 5, 1), ncol = 2, nrow = 50))
y <- as.factor(c(rep("negative", 1000), rep("positive", 50)))

plot(x, col = y)

# resampling
m <- SMOTE(x = x, y = y, k = 7)

plot(m$x_new, col = m$y_new)
```

SMOTEWB *SMOTE with boosting (SMOTEWB)*

Description

Resampling with SMOTE with boosting.

Usage

```
SMOTEWB(x, y, n_weak_classifier = 100, class_weights = NULL, k_max = NULL, ...)
```

Arguments

x	feature matrix.
y	a factor class variable with two classes.
n_weak_classifier	number of weak classifiers for boosting.
class_weights	numeric vector of length two. First number is for positive class, and second is for negative. Higher the relative weight, lesser noises for that class. By default, $2 \times n_{neg}/n$ for positive and $2 \times n_{pos}/n$ for negative class.
k_max	to increase maximum number of neighbors. Default is `ceiling(n_neg/n_pos)`.
...	additional inputs for ada::ada().

Details

SMOTEWB (Saglam & Cengiz, 2022) is a SMOTE-based oversampling method which can handle noisy data and adaptively decides the appropriate number of neighbors to link during resampling with SMOTE.

Trained model based on this method gives significantly better Matthew Correlation Coefficient scores compared to others.

Value

a list with resampled dataset.

x_new	Resampled feature matrix.
y_new	Resampled target variable.
x_syn	Generated synthetic data.
w	Boosting weights for original dataset.
k	Number of nearest neighbors for positive class samples.
C	Number of synthetic samples for each positive class samples.

Author(s)

Fatih Saglam, saglamf89@gmail.com

References

Sağlam, F., & Cengiz, M. A. (2022). A novel SMOTE-based resampling technique trough noise detection and the boosting procedure. Expert Systems with Applications, 200, 117023.

Can work with 2 classes only yet.

Examples

```
set.seed(1)
x <- rbind(matrix(rnorm(2000, 3, 1), ncol = 2, nrow = 1000),
           matrix(rnorm(100, 5, 1), ncol = 2, nrow = 50))
y <- as.factor(c(rep("negative", 1000), rep("positive", 50)))

plot(x, col = y)

# resampling
m <- SMOTEWB(x = x, y = y, n_weak_classifier = 150)

plot(m$x_new, col = m$y_new)
```

Index