Package 'smotefamily'

January 30, 2018

Title A Collection of Oversampling Techniques for Class Imbalance Problem Based on SMOTE

Version 1.2

Date 2018-01-30

Maintainer Wacharasak Siriseriwan <wacharasak.s@gmail.com>

Description A collection of various oversampling techniques developed from SMOTE is provided. SMOTE is a oversampling technique which synthesizes a new minority instance between a pair of one minority instance and one of its K nearest neighbor. (see <https://www.jair.org/media/953/live-953-2037-jair.pdf> for more information) Other techniques adopt this concept with other criteria in order to generate balanced dataset for class imbalance problem.

License GPL-2

Depends R(>= 3.0.0)

Suggests FNN, dbscan, igraph,

NeedsCompilation no

Author Wacharasak Siriseriwan [aut, cre]

Repository CRAN

Date/Publication 2018-01-30 12:55:53 UTC

R topics documented:

ADASYN *Adaptive Synthetic Sampling Approach for Imbalanced Learning*

Description

Generate synthetic positive instances using ADASYN algorithm. The number of majority neighbors of each minority instance determines the number of synthetic instances generated from the minority instance.

Usage

```
ADAS(X,target,K=5)
```

Arguments

X	A data frame or matrix of numeric-attributed dataset
target	A vector of a target class attribute corresponding to a dataset X.
K	The number of nearest neighbors during sampling process

Value

data	A resulting dataset consists of original minority instances, synthetic minority instances and original majority instances with a vector of their respective target class appended at the last column
syn_data	A set of synthetic minority instances with a vector of minority target class appended at the last column
orig_N	A set of original instances whose class is not oversampled with a vector of their target class appended at the last column
orig_P	A set of original instances whose class is oversampled with a vector of their target class appended at the last column
K	The value of parameter K for nearest neighbor process used for generating data
K_all	Unavailable for this method
dup_size	A vector of times of synthetic minority instances over original majority instances in the oversampling in each instances
outcast	Unavailable for this method
eps	Unavailable for this method
method	The name of oversampling method used for this generated dataset (ADASYN)

Author(s)

Wacharasak Siriseriwan <wacharasak.s@gmail.com>

References

He, H., Bai, Y., Garcia, E. and Li, S. 2008. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. Proceedings of IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference. pp.1322-1328.

Examples

```
data_example = sample_generator(10000,ratio = 0.80)
genData = ADAS(data_example[,-3],data_example[,3])
genData_2 = ADAS(data_example[,-3],data_example[,3],K=7)
```

ANS *Adaptive Neighbor Synthetic Majority Oversampling TEchnique*

Description

Generate a oversampling dataset from imbalanced dataset using Adaptive Neighbor SMOTE which provides the parameter K to each minority instance automatically

Usage

```
ANS(X, target, dupSize = 0)
```

Arguments

X	A data frame or matrix of numeric-attributed dataset
target	A vector of a target class attribute corresponding to a dataset X.
dupSize	A number of vector representing the desired times of synthetic minority instances over the original number of majority instances, 0 for balanced dataset.

Value

data	A resulting dataset consists of original minority instances, synthetic minority instances and original majority instances with a vector of their respective target class appended at the last column
syn_data	A set of synthetic minority instances with a vector of minority target class appended at the last column
orig_N	A set of original instances whose class is not oversampled with a vector of their target class appended at the last column
orig_P	A set of original instances whose class is oversampled with a vector of their target class appended at the last column
K	A vector of parameter K for each minority instance
K_all	The value of parameter C for nearest neighbor process used for identifying outcasts

dup_size	The maximum times of synthetic minority instances over original majority instances in the oversampling
outcast	A set of original minority instances which is defined as minority outcast
eps	The value of eps which determines automatic K
method	The name of oversampling method used for this generated dataset (ANS)

Author(s)

Wacharasak Siriseriwan <wacharasak.s@gmail.com>

References

Siriseriwan, W. and Sinapiromsaran, K. Adaptive neighbor Synthetic Minority Oversampling TEchnique under 1NN outcast handling.Songklanakarin Journal of Science and Technology.

Examples

```
data_example = sample_generator(5000,ratio = 0.80)
genData = ANS(data_example[,-3],data_example[,3])
```

Borderline-SMOTE *Borderline-SMOTE*

Description

Generate synthetic positive instances using Borderline-SMOTE algorithm. The number of majority neighbor of each minority instance is used to divide minority instances into 3 groups; SAFE/DANGER/NOISE, only the DANGER are used to generate synthetic instances.

Usage

```
BLSMOTE(X,target,K=5,C=5,dupSize=0,method =c("type1","type2"))
```

Arguments

X	A data frame or matrix of numeric-attributed dataset
target	A vector of a target class attribute corresponding to a dataset X.
K	The number of nearest neighbors during sampling process
C	The number of nearest neighbors during calculating safe-level process
dupSize	The number or vector representing the desired times of synthetic minority instances over the original number of majority instances, 0 for duplicating until balanced
method	A parameter to indicate which type of Borderline-SMOTE presented in the paper is used

Value

data	A resulting dataset consists of original minority instances, synthetic minority instances and original majority instances with a vector of their respective target class appended at the last column
syn_data	A set of synthetic minority instances with a vector of minority target class appended at the last column
orig_N	A set of original instances whose class is not oversampled with a vector of their target class appended at the last column
orig_P	A set of original instances whose class is oversampled with a vector of their target class appended at the last column
K	The value of parameter K for nearest neighbor process used for generating data
K_all	The value of parameter C for nearest neighbor process used for determining SAFE/DANGER/NOISE
dup_size	The maximum times of synthetic minority instances over original majority instances in the oversampling
outcast	Unavailable for this method
eps	Unavailable for this method
method	The name of oversampling method and type used for this generated dataset (BLSMOTE type1/2)

Author(s)

Wacharasak Siriseriwan <wacharasak.s@gmail.com>

References

Han, H., Wang, W.Y. and Mao, B.H. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In Proceedings of the 2005 international conference on Advances in Intelligent Computing - Volume Part I (ICIC'05), De-Shuang Huang, Xiao-Ping Zhang, and Guang-Bin Huang (Eds.), Vol. Part I. Springer-Verlag, Berlin, Heidelberg, 2005. 878-887. DOI=http://dx.doi.org/10.1007/11538059_91

Examples

```
    data_example = sample_generator(5000,ratio = 0.80)
genData = BLSMOTE(data_example[,-3],data_example[,3])
genData_2 = BLSMOTE(data_example[,-3],data_example[,3],K=7, C=5, method = "type2")
```

DBSMOTE *Density-based SMOTE*

Description

Generate a oversampling dataset from imbalance dataset using Density-based SMOTE. Using density reachability concept to cluster minority instances and generate synthetic instances.

Usage

```
DBSMOTE(X, target, dupSize = 0, MinPts = NULL, eps = NULL)
```

Arguments

X	A data frame or matrix of numeric-attributed dataset
target	A vector of a target class attribute
dupSize	A number of vector representing the desired times of synthetic minority instances over the original number of majority instances
MinPts	The minimum instance parameter to decide whether each instance inside eps is reachable, the automatic algorithm is used to find the value instead if there is no positive integer value given for it.
eps	The radius to consider neighbor.

Value

data	A resulting dataset consists of original minority instances, synthetic minority instances and original majority instances with a vector of their respective target class appended at the last column
syn_data	A set of synthetic minority instances with a vector of minority target class appended at the last column
orig_N	A set of original instances whose class is not oversampled with a vector of their target class appended at the last column
orig_P	A set of original instances whose class is oversampled with a vector of their target class appended at the last column
K	Unavailable for this method
K_all	Unavailable for this method
dup_size	The maximum times of synthetic minority instances over original majority instances in the oversampling
outcast	A set of original minority instances which is defined as NOISE/minority outcast
eps	The value of parameter eps
method	The name of oversampling method used for this generated dataset

Author(s)

Wacharasak Siriseriwan <wacharasak.s@gmail.com>

References

Bunkhumpornpat, C., Sinapiromsaran, K. and Lursinsap, C. 2012. DBSMOTE: Density-based synthetic minority oversampling technique. Applied Intelligence. 36, 664-684.

Examples

```
data_example = sample_generator(5000,ratio = 0.90)
genData = DBSMOTE(data_example[,-3],data_example[,3])
```

gap — *The function to provide a random number which is used as a location of synthetic instance*

Description

The function to provide a random number which uses to identify the location of each synthetic instance. The interval of possible values depends from safe-level values of instances in a pair.

Usage

```
gap(sl_p = 1, sl_n = 1)
```

Arguments

sl_p	The safe-level value of the first instance
sl_n	The safe-level value of the second instance

Value

A value between 0 to 1 which is used to identify the location of synthetic instance If sl_p >= sl_n, it gives the random number between 0 to sl_n/sl_p If sl_p < sl_n, it gives the random number between 1-sl_p/sl_n to 1

Author(s)

Wacharasak Siriseriwan <wacharasak.s@gmail.com>

Examples

```
r_num = gap()
r_num_2 = gap(sl_p = 4, sl_n = 2)
```

kncount — *Counting the number of each class in K nearest neighbor*

Description

The function to count how many neighbor of each instance belong to each class.

Usage

```
kncount(knidex, classArray)
```

Arguments

knidex	The matrix of K nearest neighbor of dataset
classArray	The index of last instance of the first class in the dataset or the vector containing indices of last instances of each class.

Details

The dataset is expected to be sorted as all m1 instances in the first class are in the first m1 instances of the dataset following with all m2 instances in the next m2 instances etc. before performing k-nearest neighbor with the knearest function.

Value

The matrix with the number of columns equal to the number of classes. Each a[i][j] represents the number of K-nearest neighbors of i th instance belonging to the class j th

Author(s)

Wacharasak Siriseriwan <wacharasak.s@gmail.com>

Examples

```
D = sample_generator(1000,ratio = 0.8)
P = D[D[,3]=="p",]
N = D[D[,3]=="n",]
D_arr=rbind(P,N)
    knear=knearest(D_arr[,-3],P[,-3],5)
kncount_result = kncount(knear,nrow(P))
```

---

knearest *The function to find n_clust nearest neighbors of each instance, always removing the index of that instance if it is reported.*

Description

The function will find n_clust nearest neighbors of each instance using Fast nearest neighbors (through KD-tree method) but will correct the result if it reports the index of that instance as its neighbors.

Usage

```
knearest(D, P, n_clust)
```

Arguments

D	a query data matrix.
P	an input data matrix
n_clust	the maximum number of nearest neighbors to search

Details

This function will perform K-nearest neighbor of instances in P on instances in P based on FNN. Then, it will verify if one of neighbors of each instance is itself then removes if it is.

Value

The index matrix of K nearest neighbour of each instance

Author(s)

Wacharasak Siriseriwan <wacharasak.s@gmail.com>

Examples

```
data_example = sample_generator(10000,ratio = 0.80)
P = data_example[data_example[,3]=="p",-3]
N = data_example[data_example[,3]=="n",-3]
D = rbind(P,N)
knear = knearest(D,P,n_clust = 5)
```

---

n_dup_max *The function to calculate the maximum round each sampling is repeated*

Description

The function to calculate the maximum round each sampling is repeated, if dup_size is given as 0 then, it calculates the maximum round the number of positive instances to be duplicated to nearly match the number of negative instances

Usage

```
n_dup_max(size_input, size_P, size_N, dup_size = 0)
```

Arguments

size_input	The size of overall dataset
size_P	The number of positive instances
size_N	The number of negative instances
dup_size	A number or vector of the number of times to be duplicated. The default is zero which means duplicating until nearly balanced.

Value

If dup_size is zero or contains zero, the number of rounds to duplicate positive to nearly equal to the number of negative instances If dup_size is not zero or contains no zero, the maximum value in dup_size

Author(s)

Wacharasak Siriseriwan <wacharasak.s@gmail.com>

Examples

```
data_example = sample_generator(10000,ratio = 0.80)
P = data_example[data_example[,3]=="p",-3]
N = data_example[data_example[,3]=="n",-3]
D = rbind(P,N)
max_round =n_dup_max(nrow(D),nrow(P),nrow(N),dup_size= 0)
```

RSLS *Relocating Safe-level SMOTE*

Description

Generate synthetic positive instances using Relocating Safe-level SMOTE algorithm. Using the parameter "Safe-Level" to determine the possible location and relocating synthetic instances if there is too close to majority instances.

Usage

```
RSLS(X, target, K = 5, C = 5, dupSize = 0)
```

Arguments

X	A data frame or matrix of numeric-attributed dataset
target	A vector of a target class attribute corresponding to a dataset X.
K	The number of nearest neighbors during sampling process
C	The number of nearest neighbors during calculating safe-level process
dupSize	The number or vector representing the desired times of synthetic minority instances over the original number of majority instances

Value

data	A resulting dataset consists of original minority instances, synthetic minority instances and original majority instances with a vector of their respective target class appended at the last column
syn_data	A set of synthetic minority instances with a vector of minority target class appended at the last column
orig_N	A set of original instances whose class is not oversampled with a vector of their target class appended at the last column
orig_P	A set of original instances whose class is oversampled with a vector of their target class appended at the last column

K	The value of parameter K for nearest neighbor process used for generating data
K_all	The value of parameter C for nearest neighbor process used for calculating safe-level
dup_size	The maximum times of synthetic minority instances over original majority instances in the oversampling
outcast	A set of original minority instances which has safe-level equal to zero and is defined as the minority outcast
eps	Unavailable for this method
method	The name of oversampling method used for this generated dataset (RSLS)

Author(s)

Wacharasak Siriseriwan <wacharasak.s@gmail.com>

References

Siriseriwan, W. and Sinapiromsaran, K. The Effective Redistribution for Imbalance Dataset : Relocating Safe-Level SMOTE with Minority Outcast Handling. Chiang Mai Journal of Science. 43(1), 234 - 246.

Examples

```
library(smotefamily)
data_example = sample_generator(5000,ratio = 0.80)
    genData = RSLS(data_example[,-3],data_example[,3])
genData_2 = RSLS(data_example[,-3],data_example[,3],K=7, C=5)
```

---

sample_generator *The function to generate 2-dimensional dataset*

Description

The function to generate 2-dimensional dataset given the number of instances and the ratio between the number of negative instances to total instances. The positive instances will be distributed uniformly as the circle in the center while negative instances are around over the domain. The random positive outcasts are also generated. The dataset is used to show the difference between datasets generated by each sampling technique.

Usage

```
sample_generator(n, ratio = 0.8, xlim = c(0, 1), ylim = c(0, 1),
   radius = 0.25, overlap = -0.05, outcast_ratio = 0.01)
```

Arguments

n	The number of instances in the dataset
ratio	The ratio of negative instances to the total number of instances
xlim	The range of values in the first dimension
ylim	The range of values in the second dimension
radius	The radius of the circle of positive instances
overlap	The gap between the set of positive and negative instances
outcast_ratio	The ratio of outcast to be generate in this dataset.

Value

A 2-dimensional dataset with the 3rd column as its target class vector.

Author(s)

Wacharasak Siriseriwan <wacharasak.s@gmail.com>

Examples

```
data_example = sample_generator(5000,ratio = 0.80)
plot(data_example[data_example[,3]=="n",1],
data_example[data_example[,3]=="n",2],col="yellow")
points(data_example[data_example[,3]=="p",1],
data_example[data_example[,3]=="p",2],col="red",pch=14)
```

SLS *Safe-level SMOTE*

Description

Generate synthetic positive instances using Safe-level SMOTE algorithm. Using the parameter "Safe-level" to determine the possible location of synthetic instances.

Usage

```
SLS(X, target, K = 5, C = 5, dupSize = 0)
```

Arguments

X	A data frame or matrix of numeric-attributed dataset
target	A vector of a target class attribute corresponding to a dataset X.
K	The number of nearest neighbors during sampling process
C	The number of nearest neighbors during calculating safe-level process
dupSize	The number or vector representing the desired times of synthetic minority instances over the original number of majority instances

Value

data	A resulting dataset consists of original minority instances, synthetic minority instances and original majority instances with a vector of their respective target class appended at the last column
syn_data	A set of synthetic minority instances with a vector of minority target class appended at the last column
orig_N	A set of original instances whose class is not oversampled with a vector of their target class appended at the last column
orig_P	A set of original instances whose class is oversampled with a vector of their target class appended at the last column
K	The value of parameter K for nearest neighbor process used for generating data
K_all	The value of parameter C for nearest neighbor process used for calculating safe-level
dup_size	The maximum times of synthetic minority instances over original majority instances in the oversampling
outcast	A set of original minority instances which has safe-level equal to zero and is defined as the minority outcast
eps	Unavailable for this method
method	The name of oversampling method used for this generated dataset (SLS)

Author(s)

Wacharasak Siriseriwan <wacharasak.s@gmail.com>

References

Bunkhumpornpat, C., Sinapiromsaran, K. and Lursinsap, C. 2009. Safe-level-SMOTE: Safe-level-synthetic minority oversampling technique for handling the class imbalanced problem. Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining. 2009, 475-482.

Examples

```
    data_example = sample_generator(5000,ratio = 0.80)
genData = SLS(data_example[,-3],data_example[,3])
genData_2 = SLS(data_example[,-3],data_example[,3],K=7, C=5)
```

SMOTE *Synthetic Minority Oversampling TEchnique*

Description

Generate synthetic positive instances using SMOTE algorithm

Usage

```
SMOTE(X, target, K = 5, dup_size = 0)
```

Arguments

X	A data frame or matrix of numeric-attributed dataset
target	A vector of a target class attribute corresponding to a dataset X.
K	The number of nearest neighbors during sampling process
dup_size	The number or vector representing the desired times of synthetic minority instances over the original number of majority instances

Value

data	A resulting dataset consists of original minority instances, synthetic minority instances and original majority instances with a vector of their respective target class appended at the last column
syn_data	A set of synthetic minority instances with a vector of minority target class appended at the last column
orig_N	A set of original instances whose class is not oversampled with a vector of their target class appended at the last column
orig_P	A set of original instances whose class is oversampled with a vector of their target class appended at the last column
K	The value of parameter K for nearest neighbor process used for generating data
K_all	Unavailable for this method
dup_size	The maximum times of synthetic minority instances over original majority instances in the oversampling
outcast	Unavailable for this method
eps	Unavailable for this method
method	The name of oversampling method used for this generated dataset (SMOTE)

Author(s)

Wacharasak Siriseriwan <wacharasak.s@gmail.com>

References

Chawla, N., Bowyer, K., Hall, L. and Kegelmeyer, W. 2002. SMOTE: Synthetic minority oversampling technique. Journal of Artificial Intelligence Research. 16, 321-357.

Examples

```
    data_example = sample_generator(10000,ratio = 0.80)
genData = SMOTE(data_example[,-3],data_example[,3])
genData_2 = SMOTE(data_example[,-3],data_example[,3],K=7)
```

Index