Package 'ROBOSRMSMOTE'

March 9, 2026

Type Package

Title Robust Oversampling with RM-SMOTE for Imbalanced Classification

Version 1.0.0

Date 2026-03-04

Description Provides the ROBOSRMSMOTE (Robust Oversampling with RM-SMOTE) framework for imbalanced
classification tasks. This package extends Mahalanobis distance-based oversampling techniques by integrating robust covariance estimators to better handle outliers and complex data distributions. The implemented methodology builds upon and significantly expands the RM-SMOTE algorithm originally proposed by Taban et al. (2025) <doi:10.1007/s10260-025-00819-8>.

License GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 4.0.0)

Imports rrcov (>= 1.7.0), meanShiftR (>= 0.56), stats

Suggests testthat (>= 3.0.0), knitr, rmarkdown

RoxygenNote 7.3.3

NeedsCompilation no

Author Emre Dunder [aut],
Mehmet Ali Cengiz [aut],
Zainab Subhi Mahmood Hawrami [aut, cre],
Abdulmohsen Alharthi [aut]

Maintainer Zainab Subhi Mahmood Hawrami <zainabsubhi@gmail.com>

Repository CRAN

Date/Publication 2026-03-09 11:20:08 UTC

Contents

get_robust_cov 2

haberman 3
ROBOS_RM_SMOTE 4
weighting 6

Index **8**

`get_robust_cov` *Get Robust Center and Covariance Matrix*

Description

Computes a robust estimate of the center (location) and covariance matrix for a given dataset using one of seven supported robust estimators.

Usage

```
get_robust_cov(data, method = "mcd")
```

Arguments

`data`	A numeric matrix or data frame containing only the feature columns (no class column). Rows are observations, columns are variables.
`method`	A character string specifying the robust covariance estimator. One of `"mcd"`, `"mve"`, `"mest"`, `"mmest"`, `"sde"`, `"sest"`, or `"ogk"`. Default is `"mcd"`.

Details

The following estimators are available via the **rrcov** package:

`mcd` Minimum Covariance Determinant (Rousseeuw & Driessen, 1999). The default and most widely used robust estimator. Suitable for most cases.

`mve` Minimum Volume Ellipsoid (Rousseeuw & Van Zomeren, 1990). An alternative to MCD, generally slower.

`mest` M-estimator of location and scatter. Iteratively re-weighted least squares approach.

`mmest` MM-estimator. Combines high breakdown point with high efficiency.

`sde` Stahel-Donoho Estimator. Projection-based robust estimator, useful for high-dimensional data.

`sest` S-estimator. High breakdown point estimator based on minimizing a robust scale.

`ogk` Orthogonalized Gnanadesikan-Kettenring estimator. Fast and stable for moderate dimensions.

Value

A list with two elements:

`center` A numeric vector of length `ncol(data)` representing the robust location estimate.

`cov` A numeric matrix of size `ncol(data)` x `ncol(data)` representing the robust covariance matrix estimate.

References

Rousseeuw, P.J. and Driessen, K.V. (1999). A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3), 212-223.

Todorov, V. and Filzmoser, P. (2009). An object-oriented framework for robust multivariate analysis. *Journal of Statistical Software*, 32(3), 1-47.

Examples

```
# Generate a simple numeric dataset
set.seed(42)
X <- matrix(rnorm(100 * 3), nrow = 100, ncol = 3)

# MCD estimator (default)
result_mcd <- get_robust_cov(X, method = "mcd")
result_mcd$center
result_mcd$cov

# OGK estimator
result_ogk <- get_robust_cov(X, method = "ogk")
result_ogk$center
```

---

haberman *Haberman Survival Imbalanced Dataset*

Description

Binary imbalanced dataset from Haberman survival study (1958-1970). Minority class represents patients who did not survive 5+ years after breast cancer surgery.

Usage

```
haberman
```

Format

A data frame with 306 rows and 4 columns:

age Age of patient at operation (numeric).

year Year of operation, 1958-1969 (numeric).

nodes Number of positive axillary nodes detected (numeric).

class `"negative"` = survived 5+ years (n=225); `"positive"` = did not survive (n=81). IR = 2.78.

Source

KEEL Repository https://sci2s.ugr.es/keel/. Used as benchmark dataset in Hawrami et al. (2025).

Examples

```
data(haberman)
table(haberman$class)
balanced <- ROBOS_RM_SMOTE(dt = haberman, target = "positive", eIR = 1)
table(balanced$class)
```

ROBOS_RM_SMOTE *RM-SMOTE: Robust Mahalanobis SMOTE for Imbalanced Classification (ROBOSRMSMOTE Framework)*

Description

Generates synthetic minority class observations using a robust version of SMOTE as part of the ROBOSRMSMOTE (Robust Oversampling with RM-SMOTE) framework. Atypical minority class observations (outliers) are down-weighted based on their robust Mahalanobis distance so that they have a lower probability of being selected as parents in the resampling step. The k-nearest neighbours of each candidate parent are also found using the robust Mahalanobis distance rather than the standard Euclidean distance.

Usage

```
ROBOS_RM_SMOTE(
  dt,
  target = "positive",
  dup_size = 0,
  eIR = 1,
  k = 5,
  threshold = 0.01,
  weight_func = 1,
  cov_method = "mcd"
)
```

Arguments

dt	A data frame containing the full (imbalanced) training set. Must include a column named "class" with exactly two distinct factor levels: "negative" (majority) and "positive" (minority). All other columns must be numeric features.
target	A character string identifying the minority class label in the "class" column. Default is "positive".
dup_size	A non-negative numeric value. When dup_size > 0, exactly round(dup_size * n_minority) synthetic observations are generated. When dup_size = 0 (default), the eIR argument controls the number of synthetic observations instead.
eIR	Expected imbalance ratio after oversampling. Used only when dup_size = 0. eIR = 1 produces a perfectly balanced dataset. eIR > 1 allows some imbalance to remain. Must satisfy 1 <= eIR < IR where IR is the original imbalance ratio. Default is 1.

k	A positive integer specifying the number of nearest neighbours used in the SMOTE resampling step. Default is 5.
threshold	A numeric value in (0, 1) passed to weighting. Controls the chi-square cutoff for outlier detection. Default is 0.01.
weight_func	An integer (1, 2, or 3) passed to weighting selecting the outlier penalisation function. Default is 1 (ω_A, hard exclusion).
cov_method	A character string passed to get_robust_cov specifying the robust covariance estimator. One of "mcd", "mve", "mest", "mmest", "sde", "sest", "ogk". Default is "mcd".

Details

The algorithm proceeds as follows (Algorithm 1 in Taban et al., 2025):

1. Extract minority class observations X_1.
2. Robustly estimate the mean vector $\hat{\mu}_1$ and covariance matrix $\hat{\Sigma}_1$ using the selected cov_method.
3. Compute the squared robust Mahalanobis distance for every minority observation.
4. Apply the selected weighting function to obtain a probability distribution Π_l over X_1.
5. Build the k-nearest neighbour graph over X_1 using the robust Mahalanobis distance.
6. Repeat until the desired number of synthetic observations is reached:
   (a) Sample the first parent x_a according to Π_l.
   (b) Choose the second parent x_b uniformly from the k neighbours of x_a.
   (c) Generate $x_{new} = v \cdot x_a + (1 - v) \cdot x_b$ where $v \sim \text{Uniform}(0, 1)$.

Value

A data frame with the same columns as dt, containing the original observations plus the newly generated synthetic minority class observations. Row names are reset to NULL.

References

Dunder, E., Cengiz, M.A., Hawrami, Z.S.M. and Alharthi, A. (2025). Robust Covariance-Based Oversampling Strategies for Imbalanced Classification. *Manuscript in preparation*.

Taban, R., Nunes, C. and Oliveira, M.R. (2025). RM-SMOTE: a new robust balancing technique. *Statistical Methods & Applications*. doi:10.1007/s10260025008198

Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16, 321-357.

See Also

get_robust_cov, weighting

Examples

```
# Load the package example dataset
data(haberman)

# Basic usage: balance with MCD (default) and hard exclusion
balanced <- ROBOS_RM_SMOTE(dt = haberman, target = "positive", eIR = 1)
table(balanced$class)

# Use MVE estimator and soft weighting (omega_B)
balanced_mve <- ROBOS_RM_SMOTE(dt = haberman, target = "positive",
                           eIR = 1, cov_method = "mve", weight_func = 2)
table(balanced_mve$class)

# Control exact number of synthetic samples with dup_size
balanced_dup <- ROBOS_RM_SMOTE(dt = haberman, target = "positive",
                           dup_size = 2, cov_method = "ogk")
table(balanced_dup$class)
```

`weighting` *Compute Robust Mahalanobis Weights for Minority Class Observations*

Description

For each minority class observation, computes the robust Mahalanobis distance (MD) to the class center and assigns a weight based on the chosen weighting function. Observations flagged as outliers (MD exceeds the chi-square threshold) receive reduced or zero weight, lowering their probability of being selected as parents in the SMOTE resampling step.

Usage

```
weighting(data, threshold = 0.01, weight_func = 1, cov_method = "mcd")
```

Arguments

`data`	A data frame of minority class observations. The last column must be the class label column named `"class"`. All other columns must be numeric features.
`threshold`	A numeric value in `(0, 1)` representing the significance level used to compute the chi-square cutoff for outlier detection. Common values: `0.001`, `0.01`, `0.025`. Default is `0.01`. A smaller value is less aggressive (fewer observations flagged as outliers).
`weight_func`	An integer (1, 2, or 3) selecting the weighting function applied to outlier observations: 1 Hard exclusion: outliers receive weight 0 (function ω_A in the paper). Maximum penalty. 2 Soft inverse: outliers receive weight $1/MD^2$ (function ω_B). Moderate penalty.

3 Scaled inverse: outliers receive weight $\tau_{1-\alpha}/MD^2$ (function ω_C). Minimum penalty.

cov_method A character string passed to get_robust_cov specifying the robust covariance estimator. One of "mcd", "mve", "mest", "mmest", "sde", "sest", "ogk". Default is "mcd".

Value

The input data frame with three additional columns appended:

MD Squared robust Mahalanobis distance for each observation.

weights Raw weight assigned to each observation (1 for non-outliers, reduced for outliers).

prob Normalised selection probability derived from weights. Sums to 1 across all rows.

References

Taban, R., Nunes, C. and Oliveira, M.R. (2025). RM-SMOTE: a new robust balancing technique. *Statistical Methods & Applications*. doi:10.1007/s10260025008198

See Also

get_robust_cov, ROBOS_RM_SMOTE

Examples

```
# Create a small imbalanced dataset
set.seed(42)
minority <- data.frame(
  x1    = c(rnorm(18), 10, 12),   # last two are outliers
  x2    = c(rnorm(18),  9, 11),
  class = "positive"
)

# Weight with hard exclusion (omega_A)
result <- weighting(minority, threshold = 0.01, weight_func = 1)
table(result$weights)   # outliers get weight 0

# Weight with soft inverse (omega_B)
result2 <- weighting(minority, threshold = 0.01, weight_func = 2)
round(result2$prob, 4)
```

Index

∗ **datasets**
haberman, 3

get_robust_cov, 2, *5*, *7*

haberman, 3

ROBOS_RM_SMOTE, 4, *7*

weighting, *5*, 6