Package ‘randomMachines’

July 23, 2025

Type Package

Title An Ensemble Modeling using Random Machines

Version 0.1.1

Description A novel ensemble method employing Support Vector Machines (SVMs) as base learners. This powerful ensemble model is designed for both classification (Ara A., et. al, 2021) <doi:10.6339/21-JDS1014>, and regression (Ara A., et. al, 2021) <doi:10.1016/j.eswa.2022.117107> problems, offering versatility and robust performance across different datasets and compared with other consolidated methods as Random Forests (Maia M, et. al, 2021) <doi:10.6339/21-JDS1025>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Imports kernlab, methods, stats

Depends R (>= 2.10)

NeedsCompilation no

Author Mateus Maia [aut, cre] (ORCID: <`https://orcid.org/0000-0001-7056-386X`>),
Anderson Ara [cte] (ORCID: <`https://orcid.org/0000-0002-1041-2768`>),
Gabriel Ribeiro [cte]

Maintainer Mateus Maia <`mateus.maiamarques@glasgow.ac.uk`>

Repository CRAN

Date/Publication 2025-07-23 13:20:10 UTC

Contents

`bolsafam` *Bolsa Família Dataset*

Description

The 'bolsafam' dataset contains information about the utilization rate of the Bolsa Família program in Brazilian municipalities. The utilization rate y_i is defined as the number of people benefiting from the assistance divided by the total population of the city.

Usage

```
data(bolsafam)
```

Format

A data frame with 5564 rows and 11 columns.

Details

This dataset includes the following columns:

y Rate of use of the social assistance program by municipality.

COD_UF Code to identify the Brazilian state to which the city belongs.

T_DENS Percentage of the population living in households with a density greater than 2 people per bedroom.

TRABSC Percentage of employed persons aged 18 or over who are employed without a formal contract.

PPOB Proportion of people vulnerable to poverty.

T_NESTUDA_NTRAB_MMEIO Percentage of people aged 15 to 24 who do not study or work and are vulnerable to poverty.

T_FUND15A17 Percentage of the population aged 15 to 17 with complete primary education.

RAZDEP Dependency ratio.

T_ATRASO_0_BASICO Percentage of the population aged 6 to 17 years attending basic education that does not have an age-grade delay.

T_AGUA Percentage of the population living in households with running water.

REGIAO Aggregation of states according to the regions defined by IBGE.

Source

The 'bolsafam' dataset is sourced from the Brazilian organizational site called *Transparency Portal*.

References

Mateus Maia & Anderson Ara (2023). rmachines: Random Machines: a package for a support vector ensemble based on random kernel space. R package version 0.1.0.

Examples

```
data(bolsafam)
head(bolsafam)
```

`brier_score` *Brier Score function*

Description

Calculate the Brier Score for a set of predicted probabilities and observed outcomes. The Brier Score is a measure of the accuracy of probabilistic predictions. It is commonly used in the evaluation of predictive models.

Usage

```
brier_score(prob, observed, levels)
```

Arguments

`prob`	predicted probabilities
`observed`	y observed values (it assumed that the positive class is coded is equal to one and the negative 0)
`levels`	A string vector with the original levels from the target variable

Value

Returns the Brier Score, a numeric value indicating the accuracy of the predictions.

`ionosphere` *Ionosphere Dataset*

Description

The 'ionosphere' dataset contains radar data for the classification of radar returns as either 'good' or 'bad'.

Usage

```
data(ionosphere)
```

Format

A data frame with 351 rows and 35 columns.

Details

This dataset includes the following columns:

X1-X34 Features extracted from radar signals.

y Class label indicating whether the radar return is 'g' (good) or 'b' (bad).

Source

The 'ionosphere' dataset is sourced from the UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/ionosphere

Examples

```
data(ionosphere)
head(ionosphere)
```

`predict.rm_class` *Prediction function for the rm_class_model*

Description

This function predicts the outcome for a RM object model using new data

Usage

```
## S4 method for signature 'rm_class'
predict(object,newdata)
```

Arguments

`object`	A fitted RM model object of class `rm_class`.
`newdata`	A data frame or matrix containing the new data to be predicted.

Value

A vector of predicted outcomes: probabilities in case of ‘prob_model = TRUE‘ and classes in case of ‘prob_model = FALSE‘.

Examples

```
# Generating a sample for the simulation
library(randomMachines)
sim_data <- sim_class(n = 75)
sim_new <- sim_class(n = 25)
rm_mod <- randomMachines(y~., train = sim_data)
y_hat <- predict(rm_mod, newdata = sim_new)
```

`predict.rm_reg` *Prediction function for the rm_reg_model*

Description

This function predicts the outcome for a RM object model using new data for continuous y

Usage

```
## S4 method for signature 'rm_reg'
predict(object,newdata)
```

Arguments

`object`	A fitted RM model object of class `rm_reg`.
`newdata`	A data frame or matrix containing the new data to be predicted.

Value

Predicted values `newdata` object from the Random Machines model.

Examples

```
# Generating a sample for the simulation
library(randomMachines)
sim_data <- sim_reg1(n = 75)
sim_new <- sim_reg1(n = 25)
rm_mod_reg <- randomMachines(y~., train = sim_data)
y_hat <- predict(rm_mod_reg, newdata = sim_new)
```

`randomMachines` *Random Machines*

Description

Random Machines is an ensemble model which uses the combination of different kernel functions to improve the diversity in the bagging approach, improving the predictions in general. Random Machines was developed for classification and regression problems by bagging multiple kernel functions in support vector models.

Random Machines uses SVMs (Cortes and Vapnik, 1995) as base learners in the bagging procedure with a random sample of kernel functions to build them.

Let a training sample given by $(\boldsymbol{x_i}, y_i)$ with $i = 1, \ldots, n$ observations, where $\boldsymbol{x_i}$ is the vector of independent variables and y_i the dependent one. The kernel bagging method initializes by training of the r single learner, where $r = 1, \ldots, R$ and R is the total number of different kernel functions that could be used in support vector models. In this implementation the default value is $R = 4$ (gaussian, polynomial, laplacian and linear). See more details below.

Each single learner is internally validated and the weights λ_r are calculated proportionally to the strength from the single predictive performance.

Afterwards, B bootstrap samples are sampled from the training set. A support vector machine model g_b is trained for each bootstrap sample, $b = i, \ldots, B$ and the kernel function that will be used for g_b will be determined by a random choice with probability λ_r. The final weight w_b in the bagging procedure is calculated by out-of-bag samples.

The final model $G(\boldsymbol{x}_i)$ for a new $\boldsymbol{x}_i$ is given by,

The weights λ_r and w_b are different calculated for each task (classification, probabilistic classification and regression). See more details in the references.

- For a binary classification problem $G(\boldsymbol{x_i}) = \text{sgn}\left(\sum_{b=1}^{B} w_b g_b(\boldsymbol{x_i})\right)$, where g_b are single binary classification outputs;
- For a probabilistic binary classification problem $G(\boldsymbol{x_i}) = \sum_{b=1}^{B} w_b g_b(\boldsymbol{x_i})$, where g_b are single probabilistic classification outputs;
- For a regression problem $G(\boldsymbol{x_i}) = \sum_{b=1}^{B} w_b g_b(\boldsymbol{x_i})$, , where g_b are single regression outputs.

Usage

```
randomMachines(
     formula,
     train,validation,
     B = 25, cost = 1,
     automatic_tuning = FALSE,
     gamma_rbf = 1,
     gamma_lap = 1,
     degree = 2,
     poly_scale = 1,
     offset = 0,
```

```
      gamma_cau = 1,
      d_t = 2,
      kernels = c("rbfdot", "polydot", "laplacedot", "vanilladot"),
      prob_model = TRUE,
      loss_function = RMSE,
      epsilon = 0.1,
      beta = 2
)
```

Arguments

`formula` an object of class `formula`: it should contain a symbolic description of the model to be fitted, indicating the dependent variable and all predictors that should be included.

`train` the training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ used to train the model.

`validation` the validation data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{V}$ used to calculate probabilities λ_r. If `validation = NULL`,the validation set is going be selected as 0.25 partition from the training data, and the remaining partition is selected as the new training sample.

`B` number of bootstrap samples. The default value is `B=25`.

`cost` the C-constant term of the regularization on soft margins at support vector models. The default value is `cost=1`.

`automatic_tuning` boolean to define if the kernel hyperparameters will be selected using the `sigest` from the `ksvm` function. The default value is `FALSE`.

`gamma_rbf` the hyperparameter γ_g used in the RBF kernel. The default value is `gamma_rbf=1`.

`gamma_lap` the hyperparameter γ_l used in the Laplacian kernel. The default value is `gamma_lap=1`.

`degree` the degree used in the Polynomial kernel. The default value is `degree=2`.

`poly_scale` the scale parameter from the Polynomial kernel. The default value is `poly_scale=1`.

`offset` the offset parameter from the Polynomial kernel. The default value is `offset=0`.

`gamma_cau` the hyperparameter γ_c used in the Cauchy kernel. The default value is `gamma_cau=1`.

`d_t` the d_t-norm from the t-Student kernel. The default value is `d_t=2`.

`kernels` a vector with the name of kernel functions that will be used in the Random Machines model. The default include the kernel functions: `c("rbfdot", "polydot", "laplacedot", "vanilladot")`. The other kernel functions as `"cauchydot"` and `"tdot"` are exclusive to the binary classification setting.

`prob_model` a boolean to define if the algorithm will be using a probabilistic approach to the define the predictions (default = `TRUE`).

`loss_function` Define which loss function is going to be used in the regression approach. The default is the `RMSE` function but others can be added.

`epsilon` The epsilon in the loss function used from the SVR implementation. The default value is `epsilon=0.1`.

`beta` The correlation parameter β which calibrates the penalisation of each kernel performance in regression tasks. The default value is `beta=2`.

Details

The Random Machines is an ensemble method which combines the bagging procedure proposed by Breiman (1996), using Support Vector Machine models as base learners jointly with a random selection of kernel functions that add diversity to the ensemble without harming its predictive performance. The kernel functions $k(x, y)$ are described by the functions below,

- Linear Kernel: $k(x, y) = (x \cdot y)$
- Polynomial Kernel: $k(x, y) = (scale(x \cdot y) + offset)^{degree}$
- Gaussian Kernel: $k(x, y) = e^{-\gamma_g ||x-y||^2}$
- Laplacian Kernel: $k(x, y) = e^{-\gamma_\ell ||x-y||}$
- Cauchy Kernel: $k(x, y) = \frac{1}{1+\frac{||x-y||^2}{\gamma_c}}$
- Student's t Kernel: $k(x, y) = \frac{1}{1+||x-y||^{d_t}}$

Value

`randomMachines()` returns an object of `class` "rm_class" for classification tasks or "rm_reg" for if the target variable is a continuous numerical response. See `predict.rm_class` or `predict.rm_reg` for more details of how to obtain predictions from each model respectively.

Author(s)

Mateus Maia: <mateusmaia11@gmail.com>, Gabriel Felipe Ribeiro: <brielribeiro08@gmail.com>, Anderson Ara: <ara@ufpr.br>

References

Ara, Anderson, et al. "Regression random machines: An ensemble support vector regression model with free kernel choice." Expert Systems with Applications 202 (2022): 117107.

Ara, Anderson, et al. "Random machines: A bagged-weighted support vector model with free kernel choice." Journal of Data Science 19.3 (2021): 409-428.

Breiman, L. (1996). Bagging predictors. Machine learning, 24, 123-140.

Cortes, C., and Vapnik, V. (1995). Support-vector networks. Machine learning, 20, 273-297.

Maia, Mateus, Arthur R. Azevedo, and Anderson Ara. "Predictive comparison between random machines and random forests." Journal of Data Science 19.4 (2021): 593-614.

Examples

```
library(randomMachines)

# Simulation from a binary output context
sim_data <- sim_class(n = 75)

## Setting the training and validation set
sim_new <- sim_class(n = 75)

# Modelling Random Machines (probabilistic output)
```

```
rm_mod_prob <- randomMachines(y~., train = sim_data)

## Modelling Random Machines (binary class output)
rm_mod_label <- randomMachines(y~., train = sim_data,prob_model = FALSE)

## Predicting for new data
y_hat <- predict(rm_mod_label,sim_new)
```

RMSE *Root Mean Squared Error (RMSE) Function*

Description

Computes the Root Mean Squared Error (RMSE), a widely used metric for evaluating the accuracy of predictions in regression tasks. The formula is given by

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Usage

```
RMSE(predicted, observed)
```

Arguments

`predicted`	A vector of predicted values $\hat{\mathbf{y}}$.
`observed`	A vector of observed values $\mathbf{y}$.

Value

a the Root Mean Squared error calculated by the formula in the description.

---

rm_class-class *S4 class for RM classification*

Description

S4 class for RM classification

Details

For more details see Ara, Anderson, et al. "Random machines: A bagged-weighted support vector model with free kernel choice." Journal of Data Science 19.3 (2021): 409-428.

Slots

- `train` a `data.frame` corresponding to the training data used into the model
- `class_name` a string with target variable used in the model
- `kernel_weight` a numeric vector corresponding to the weights for each bootstrap model contribution
- `lambda_values` a named list with value of the vector of $\boldsymbol{\lambda}$ sampling probabilities associated with each each kernel function
- `model_params` a list with all used model specifications
- `bootstrap_models` a list with all `ksvm` objects for each bootstrap sample
- `bootstrap_samples` a list with all bootstrap samples used to train each base model of the ensemble
- `prob` a boolean indicating if a probabilitistic approch was used in the classification Random Machines

---

`rm_reg-class` *S4 class for RM regression*

Description

S4 class for RM regression

Details

For more details see Ara, Anderson, et al. "Regression random machines: An ensemble support vector regression model with free kernel choice." Expert Systems with Applications 202 (2022): 117107.

Slots

- `y_train_hat` a numeric corresponding to the predictions $\hat{y}_i$ for the training set
- `lambda_values` a named list with value of the vector of $\boldsymbol{\lambda}$ sampling probabilities associated with each each kernel function
- `model_params` a list with all used model specifications
- `bootstrap_models` a list with all `ksvm` objects for each bootstrap sample
- `bootstrap_samples` a list with all bootstrap samples used to train each base model of the ensemble
- `kernel_weight_norm` a numeric vector corresponding to the normalised weights for each bootstrap model contribution

sim_class *Generate a binary classification data set from normal distribution*

Description

Simulation used as example of a classification task based on a separation of two normal multivariate distributions with different vector of means and differerent covariate matrices. For the label A the $\mathbf{X}_A$ are sampled from a normal distribution $MVN\left(\mu_A \mathbf{1}_p, \sigma_A^2 \mathbf{I}_p\right)$ while for label B the samples $\mathbf{X}_B$ are from a normal distribution $MVN\left(\mu_B \mathbf{1}_p, \sigma_B^2 \mathbf{I}_p\right)$. For more details see Ara *et. al* (2021), and Breiman L (1998).

Usage

```
sim_class(
  n,
  p = 2,
  ratio = 0.5,
  mu_a = 0,
  sigma_a = 1,
  mu_b = 1,
  sigma_b = 1
)
```

Arguments

n	Sample size
p	Number of predictors
ratio	Ratio between class A and class B
mu_a	Mean of X_1.
sigma_a	Standard deviation of X_1.
mu_b	Mean of X_2
sigma_b	Standard devation of X_2

Value

A simulated data.frame with two predictors for a binary classification problem

Author(s)

Mateus Maia: <mateusmaia11@gmail.com>, Anderson Ara: <ara@ufpr.br>

References

Ara, Anderson, et al. "Random machines: A bagged-weighted support vector model with free kernel choice." Journal of Data Science 19.3 (2021): 409-428.

Breiman, L. (1998). Arcing classifier (with discussion and a rejoinder by the author). The annals of statistics, 26(3), 801-849.

Examples

```
library(randomMachines)
sim_data <- sim_class(n = 100)
```

`sim_reg1` *Simulation for a regression toy examples from Random Machines Regression 1*

Description

Simulation toy example initially found in Scornet (2016), and used and escribed by Ara *et. al* (2022). Inputs are 2 independent variables uniformly distributed on the interval $[-1, 1]$. Outputs are generated following the equation

$$Y = X_1^2 + e^{-X_2^2} + \varepsilon, \varepsilon \sim \mathcal{N}(0, \sigma^2)$$

Usage

```
sim_reg1(n, sigma)
```

Arguments

`n`	Sample size
`sigma`	Standard deviation of residual noise

Value

A simulated data.frame with two predictors and the target variable.

Author(s)

Mateus Maia: `<mateusmaia11@gmail.com>`, Anderson Ara: `<ara@ufpr.br>`

References

Ara, Anderson, et al. "Regression random machines: An ensemble support vector regression model with free kernel choice." Expert Systems with Applications 202 (2022): 117107.

Scornet, E. (2016). Random forests and kernel methods. IEEE Transactions on Information Theory, 62(3), 1485-1500.

Examples

```
library(randomMachines)
sim_data <- sim_reg1(n=100)
```

`sim_reg2` *Simulation for a regression toy examples from Random Machines Regression 2*

Description

Simulation toy example initially found in Scornet (2016), and used and escribed by Ara *et. al* (2022). Inputs are 8 independent variables uniformly distributed on the interval $[-1, 1]$. Outputs are generated following the equation

$$Y = X_1X_2 + X_3^2 - X_4X_7 + X_5X_8 - X_6^2 + \varepsilon, \varepsilon \sim \mathcal{N}(0, \sigma^2)$$

Usage

```
sim_reg2(n, sigma)
```

Arguments

`n`	Sample size
`sigma`	Standard deviation of residual noise

Value

A simulated data.frame with two predictors and the target variable.

Author(s)

Mateus Maia: `<mateusmaia11@gmail.com>`, Anderson Ara: `<ara@ufpr.br>`

References

Ara, Anderson, et al. "Regression random machines: An ensemble support vector regression model with free kernel choice." Expert Systems with Applications 202 (2022): 117107.

Scornet, E. (2016). Random forests and kernel methods. IEEE Transactions on Information Theory, 62(3), 1485-1500.

Examples

```
library(randomMachines)
sim_data <- sim_reg2(n=100)
```

`sim_reg3` *Simulation for a regression toy examples from Random Machines Regression 3*

Description

Simulation toy example initially found in Scornet (2016), and used and escribed by Ara *et. al* (2022). Inputs are 4 independent variables uniformly distributed on the interval $[-1, 1]$. Outputs are generated following the equation

$$Y = -\sin(X_1) + X_2^2 + X_3 - e^{-X_4^2} + \varepsilon, \varepsilon \sim \mathcal{N}(0, 0.5)$$

Usage

```
sim_reg3(n, sigma)
```

Arguments

`n`	Sample size
`sigma`	Standard deviation of residual noise

Value

A simulated data.frame with two predictors and the target variable.

Author(s)

Mateus Maia: <`mateusmaia11@gmail.com`>, Anderson Ara: <`ara@ufpr.br`>

References

Ara, Anderson, et al. "Regression random machines: An ensemble support vector regression model with free kernel choice." Expert Systems with Applications 202 (2022): 117107.

Scornet, E. (2016). Random forests and kernel methods. IEEE Transactions on Information Theory, 62(3), 1485-1500.

Examples

```
library(randomMachines)
sim_data <- sim_reg3(n=100)
```

`sim_reg4` *Simulation for a regression toy examples from Random Machines Regression 3*

Description

Simulation toy example initially found in Van der Laan, *et.al* (2016), and used and escribed by Ara *et. al* (2022). Inputs are 6 independent variables uniformly distributed on the interval $[-1, 1]$. Outputs are generated following the equation

$$Y = X_1^2 + X_2^2 X_3 e^{-|X_4|} + X_6 - X_5 + \varepsilon, \varepsilon \sim \mathcal{N}(0, \sigma^2)$$

Usage

```
sim_reg4(n, sigma)
```

Arguments

`n`	Sample size
`sigma`	Standard deviation of residual noise

Value

A simulated data.frame with two predictors and the target variable.

Author(s)

Mateus Maia: `<mateusmaia11@gmail.com>`, Anderson Ara: `<ara@ufpr.br>`

References

Ara, Anderson, et al. "Regression random machines: An ensemble support vector regression model with free kernel choice." Expert Systems with Applications 202 (2022): 117107.

Van der Laan, M. J., Polley, E. C., & Hubbard, A. E. (2007). Super learner. Statistical applications in genetics and molecular biology, 6(1).

Examples

```
library(randomMachines)
sim_data <- sim_reg4(n=100)
```

`sim_reg5` *Simulation for a regression toy examples from Random Machines Regression 3*

Description

Simulation toy example initially found in Van der Laan, *et.al* (2016), and used and escribed by Ara *et. al* (2022). Inputs are 6 independent variables sampled from $N(0,1)$. Outputs are generated following the equation

$$Y = X_1 + 0.707X_2^2 + 2\infty_{(X_3>0)} + 0.873\log(X_1)|X_3| + 0.894X_2X_4 + 2\infty_{(X_5>0)} + 0.464e^{X_6} + \varepsilon, \varepsilon \sim \mathcal{N}(0,\sigma^2)$$

Usage

```
sim_reg5(n, sigma)
```

Arguments

`n`	Sample size
`sigma`	Standard deviation of residual noise

Value

A simulated data.frame with two predictors and the target variable.

Author(s)

Mateus Maia: <mateusmaia11@gmail.com>, Anderson Ara: <ara@ufpr.br>

References

Ara, Anderson, et al. "Regression random machines: An ensemble support vector regression model with free kernel choice." Expert Systems with Applications 202 (2022): 117107.

Roy, M. H., & Larocque, D. (2012). Robustness of random forests for regression. Journal of Nonparametric Statistics, 24(4), 993-1006.

Examples

```
library(randomMachines)
sim_data <- sim_reg5(n=100)
```

`whosale` *Wholesale Dataset*

Description

The 'whosale' dataset contains information about wholesale customers' annual spending on various product categories.

Usage

```
data(whosale)
```

Format

A data frame with 440 rows and 8 columns.

Details

This dataset includes the following columns:

y Type of customer, either 'Horeca' (Hotel/Restaurant/Cafe), coded as `1` or 'Retail' coded as `2`.

Region Geographic region of the customer, either 'Lisbon', 'Oporto', or 'Other'. Coded as `{1,2,3}`, respectively.

Fresh Annual spending (in monetary units) on fresh products.

Milk Annual spending on milk products.

Grocery Annual spending on grocery products.

Frozen Annual spending on frozen products.

Detergents Paper Annual spending on detergents and paper products.

Delicassen Annual spending on delicatessen products.

Source

The 'whosale' dataset is sourced from the UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/wholesale+customers

Examples

```
data(whosale)
head(whosale)
```

Index