

Package ‘RaSEn’

January 15, 2021

Type Package

Title Random Subspace Ensemble Classification and Variable Screening

Version 2.0.0

Author Ye Tian [aut, cre] and Yang Feng [aut]

Maintainer Ye Tian <ye.t@columbia.edu>

Description We propose a general ensemble classification framework, RaSE algorithm, for the sparse classification problem. In RaSE algorithm, for each weak learner, some random subspaces are generated and the optimal one is chosen to train the model on the basis of some criterion. To be adapted to the problem, a novel criterion, ratio information criterion (RIC) is put up with based on Kullback-Leibler divergence. Besides minimizing RIC, multiple criteria can be applied, for instance, minimizing extended Bayesian information criterion (eBIC), minimizing training error, minimizing the validation error, minimizing the cross-validation error, minimizing leave-one-out error. There are various choices of base classifier, for instance, linear discriminant analysis, quadratic discriminant analysis, k-nearest neighbour, logistic regression, decision trees, random forest, support vector machines. RaSE algorithm can also be applied to do feature ranking, providing us the importance of each feature based on the selected percentage in multiple subspaces. RaSE framework can be extended to the general prediction framework, including both classification and regression. We can use the selected percentages of variables for variable screening. The latest version added the variable screening function available for both regression and classification.

Imports MASS, caret, class, doParallel, e1071, foreach, nnet, randomForest, rpart, stats, ggplot2, gridExtra, formatR, FNN, ranger

License GPL-2

Encoding UTF-8

LazyData TRUE

RoxygenNote 7.1.0

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2021-01-15 06:20:03 UTC

R topics documented:

predict.RaSE	2
print.RaSE	3
RaModel	4
RaPlot	5
RaRank	6
RaScreen	7
Rase	11

Index	17
--------------	-----------

predict.RaSE	<i>Predict the outcome of new observations based on the estimated RaSE classifier.</i>
--------------	--

Description

Predict the outcome of new observations based on the estimated RaSE classifier.

Usage

```
## S3 method for class 'RaSE'
predict(object, newx, ...)
```

Arguments

object	fitted 'RaSE' object using Rase.
newx	a set of new observations. Each row of newx is a new observation.
...	additional arguments.

Value

The predicted labels for new observations.

See Also

[Rase](#).

Examples

```
## Not run:
set.seed(0, kind = "L'Ecuyer-CMRG")
train.data <- RaModel(1, n = 100, p = 50)
test.data <- RaModel(1, n = 100, p = 50)
xtrain <- train.data$x
ytrain <- train.data$y
xtest <- test.data$x
ytest <- test.data$y
```

```
model.fit <- RaSE(xtrain, ytrain, B1 = 100, B2 = 100, iteration = 0, cutoff = TRUE,
base = 'lda', cores = 2, criterion = 'ric', ranking = TRUE)
ypred <- predict(model.fit, xtest)

## End(Not run)
```

print.RaSE	<i>Print a fitted RaSE object.</i>
------------	------------------------------------

Description

Similar to the usual print methods, this function summarizes results. from a fitted 'RaSE' object.

Usage

```
## S3 method for class 'RaSE'
print(x, ...)
```

Arguments

x	fitted 'RaSE' model object.
...	additional arguments.

Value

No value is returned.

See Also

[RaSE](#).

Examples

```
set.seed(0, kind = "L'Ecuyer-CMRG")
train.data <- RaModel("classification", 1, n = 100, p = 50)
xtrain <- train.data$x
ytrain <- train.data$y

# test RaSE classifier with LDA base classifier
fit <- RaSE(xtrain, ytrain, B1 = 50, B2 = 50, iteration = 0, cutoff = TRUE,
base = 'lda', cores = 2, criterion = 'ric', ranking = TRUE)

# print the summarized results
print(fit)
```

RaModel *Generate data (x, y) from various models in two papers.*

Description

RaModel generates data from 4 models described in Tian, Y. and Feng, Y., 2020 and 6 models described in Tian, Y. and Feng, Y., 2021.

Usage

```
RaModel(model.type, model.no, n, p, p0 = 1/2, sparse = TRUE)
```

Arguments

<code>model.type</code>	indicator of the paper covering the model, which can be 'classification' (Tian, Y. and Feng, Y., 2020) or 'screening' (Tian, Y. and Feng, Y., 2021).
<code>model.no</code>	model number. It can be 1-4 when <code>model.type = 'classification'</code> and 1-6 when <code>model.type = 'screening'</code> , respectively.
<code>n</code>	sample size
<code>p</code>	data dimension
<code>p0</code>	marginal probability of class 0. Default = 0.5. Only used when <code>model.type = 'classification'</code> and <code>model.no = 1, 2, 3</code> .
<code>sparse</code>	a logistic object indicating model sparsity. Default = TRUE. Only used when <code>model.type = 'classification'</code> and <code>model.no = 1, 4</code> .

Value

<code>x</code>	$n * p$ matrix. n observations and p features.
<code>y</code>	n responses.

Note

When `model.type = 'classification'` and `sparse = TRUE`, models 1, 2, 4 require $p \geq 5$ and model 3 requires $p \geq 50$. When `model.type = 'classification'` and `sparse = FALSE`, models 1 and 4 require $p \geq 50$ and $p \geq 30$, respectively. When `model.type = 'screening'`, models 1, 4 and 5 require $p \geq 4$. Model 2 require $p \geq 5$ and model 3 require $p \geq 22$. Model 5 requires $p \geq 2$.

References

Tian, Y. and Feng, Y., 2021. RaSE: A Variable Screening Framework via Random Subspace Ensembles.

Tian, Y. and Feng, Y., 2021. RaSE: Random subspace ensemble classification. Journal of Machine Learning Research, 22, to appear.

See Also

[Rase](#), [RaScreen](#).

Examples

```
train.data <- RaModel("classification", 1, n = 100, p = 50)
xtrain <- train.data$x
ytrain <- train.data$y

## Not run:
train.data <- RaModel("screening", 2, n = 100, p = 50)
xtrain <- train.data$x
ytrain <- train.data$y

## End(Not run)
```

RaPlot

Visualize the feature ranking results of a fitted RaSE object.

Description

This function plots the feature ranking results from a fitted 'RaSE' object via ggplot2. In the figure, x-axis represents the feature number and y-axis represents the selected percentage of each feature in B1 subspaces.

Usage

```
RaPlot(
  object,
  main = NULL,
  xlab = "feature",
  ylab = "selected percentage",
  ...
)
```

Arguments

<code>object</code>	fitted 'RaSE' model object.
<code>main</code>	title of the plot. Default = NULL, which makes the title following the form 'RaSE-base' with subscript i (rounds of iterations), where base represents the type of base classifier. i is omitted when it is zero.
<code>xlab</code>	the label of x-axis. Default = 'feature'.
<code>ylab</code>	the label of y-axis. Default = 'selected percentage'.
<code>...</code>	additional arguments.

Value

a 'ggplot' object.

See Also

[Rase](#).

Examples

```
set.seed(0, kind = "L'Ecuyer-CMRG")
train.data <- RaModel("classification", 1, n = 100, p = 50)
xtrain <- train.data$x
ytrain <- train.data$y

# fit RaSE classifier with QDA base classifier
fit <- Rase(xtrain, ytrain, B1 = 50, B2 = 50, iteration = 1, base = 'qda',
            cores = 2, criterion = 'ric')

# plot the selected percentage of each feature appearing in B1 subspaces
RaPlot(fit)
```

RaRank

Rank the features by selected percentages provided by the output from RaScreen.

Description

Rank the features by selected percentages provided by the output from RaScreen.

Usage

```
RaRank(object, selected.num = "all positive", iteration = object$iteration)
```

Arguments

object	output from RaScreen.
selected.num	the number of selected variables. User can either choose from the following popular options or input an positive integer no larger than the dimension. <ul style="list-style-type: none"> • 'all positive': the number of variables with positive selected percentage. • 'D': floor(D), where D is the maximum of random subspace size. • '1.5D': floor(1.5D). • '2D': floor(2D). • '3D': floor(3D). • 'n/logn': floor(n/logn), where n is the sample size. • '1.5n/logn': floor(1.5n/logn).

- '2n/logn': floor(2n/logn).
- '3n/logn': floor(3n/logn).
- 'n-1': the sample size n - 1.
- 'p': the dimension p.

iteration indicates results from which iteration to use. It should be an positive integer. Default = the maximal iteration round used by the output from RaScreen.

Value

Selected variables (indexes).

References

Tian, Y. and Feng, Y., 2021. RaSE: Random subspace ensemble classification. Journal of Machine Learning Research, 22, to appear.

Examples

```
## Not run:
set.seed(0, kind = "L'Ecuyer-CMRG")
train.data <- RaModel("screening", 1, n = 100, p = 100)
xtrain <- train.data$x
ytrain <- train.data$y

# test RaSE screening with linear regression model and BIC
fit <- RaScreen(xtrain, ytrain, B1 = 100, B2 = 50, iteration = 0, model = 'lm',
cores = 2, criterion = 'bic')

# Select n/logn variables
RaRank(fit, selected.num = "n/logn")

## End(Not run)
```

RaScreen

Variable screening via RaSE.

Description

RaSE is a general framework for variable screening. In RaSE screening, to select each of the B1 subspaces, B2 random subspaces are generated and the optimal one is chosen according to some criterion. Then the selected proportions (equivalently, percentages) of variables in the B1 subspaces are used as importance measure to rank these variables.

Usage

```

RaScreen(
  xtrain,
  ytrain,
  xval = NULL,
  yval = NULL,
  B1 = 200,
  B2 = floor(ncol(xtrain)^1.1),
  D = NULL,
  dist = NULL,
  model = NULL,
  criterion = NULL,
  k = 5,
  cores = 1,
  seed = NULL,
  iteration = 0,
  cv = 5,
  scale = FALSE,
  C0 = 0.1,
  kl.k = NULL,
  classification = NULL,
  ...
)

```

Arguments

<code>xtrain</code>	<code>n * p</code> observation matrix. <code>n</code> observations, <code>p</code> features.
<code>ytrain</code>	<code>n</code> 0/1 observatons.
<code>xval</code>	observation matrix for validation. Default = NULL. Useful only when <code>criterion = 'validation'</code> .
<code>yval</code>	0/1 observation for validation. Default = NULL. Useful only when <code>criterion = 'validation'</code> .
<code>B1</code>	the number of weak learners. Default = 200.
<code>B2</code>	the number of subspace candidates generated for each weak learner. Default = $\text{floor}(p^{1.1})$.
<code>D</code>	the maximal subspace size when generating random subspaces. Default = NULL. It means that $D = \min(\sqrt{n}0, \sqrt{n}1, p)$ when <code>model = 'qda'</code> , $D = \min(n^{0.49}, p)$ when <code>model = 'lm'</code> , and $D = \min(\sqrt{n}, p)$ otherwise.
<code>dist</code>	the distribution for features when generating random subspaces. Default = NULL, which represents the hierarchical uniform distribution. First generate an integer d from $1, \dots, D$ uniformly, then uniformly generate a subset with cardinality d .
<code>model</code>	the model to use. Default = 'lda' when <code>classification = TRUE</code> and 'lm' when <code>classification = FALSE</code> . <ul style="list-style-type: none"> • <code>lm</code>: linear regression. Only available for regression. • <code>lda</code>: linear discriminant analysis. <code>lda</code> in MASS package. Only available for classification.

	<ul style="list-style-type: none"> • qda: quadratic discriminant analysis. qda in MASS package. Only available for classification. • knn: k-nearest neighbor. knn, knn.cv in class package, knn3 in caret package and knnreg in caret package. • logistic: logistic regression. glmnet in glmnet package. Only available for classification. • tree: decision tree. rpart in rpart package. Only available for classification. • svm: support vector machine. If kernel is not identified by user, it will use RBF kernel. svm in e1071 package. • randomforest: random forest. randomForest in randomForest package and ranger in ranger package.
critterion	<p>the criterion to choose the best subspace. Default = 'ric' when model = 'lda', 'qda'; default = 'bic' when model = 'lm' or 'logistic'; default = 'loo' when model = 'knn'; default = 'cv' and set cv = 5 when model = 'tree', 'svm', 'randomforest'.</p> <ul style="list-style-type: none"> • ric: minimizing ratio information criterion (RIC) with parametric estimation (Tian, Y. and Feng, Y., 2020). Available for binary classification and model = 'lda', 'qda', or 'logistic'. • nric: minimizing ratio information criterion (RIC) with non-parametric estimation (Tian, Y. and Feng, Y., 2020;). Available for binary classification and model = 'lda', 'qda', or 'logistic'. • training: minimizing training error/MSE. Not available when model = 'knn'. • loo: minimizing leave-one-out error/MSE. Only available when model = 'knn'. • validation: minimizing validation error/MSE based on the validation data. • cv: minimizing k-fold cross-validation error/MSE. k equals to the value of cv. Default = 5. • aic: minimizing Akaike information criterion (Akaike, H., 1973). Available when base = 'lm' or 'logistic'. $AIC = -2 * \log\text{-likelihood} + S * 2.$ • bic: minimizing Bayesian information criterion (Schwarz, G., 1978). Available when model = 'lm' or 'logistic'. $BIC = -2 * \log\text{-likelihood} + S * \log(n).$ • ebic: minimizing extended Bayesian information criterion (Chen, J. and Chen, Z., 2008; 2012). gam value is needed. When gam = 0, it represents BIC. Available when model = 'lm' or 'logistic'. $eBIC = -2 * \log\text{-likelihood} + S * \log(n) + 2 * S * \text{gam} * \log(p).$
k	the number of nearest neighbors considered when model = 'knn'. Only useful when model = 'knn'. Default = 5.
cores	the number of cores used for parallel computing. Default = 1.
seed	the random seed assigned at the start of the algorithm, which can be a real number or NULL. Default = NULL, in which case no random seed will be set.
iteration	the number of iterations. Default = 0.

<code>cv</code>	the number of cross-validations used. Default = 5. Only useful when <code>criterion = 'cv'</code> .
<code>scale</code>	whether to normalize the data. Logistic, default = FALSE.
<code>C0</code>	a positive constant used when <code>iteration > 1</code> . See Tian, Y. and Feng, Y., 2021 for details. Default = 0.1.
<code>k1.k</code>	the number of nearest neighbors used to estimate RIC in a non-parametric way. Default = NULL, which means that $k_0 = \text{floor}(\sqrt{n_0})$ and $k_1 = \text{floor}(\sqrt{n_1})$. See Tian, Y. and Feng, Y., 2020 for details. Only available when <code>criterion = 'nric'</code> .
<code>classification</code>	the indicator of the problem type, which can be TRUE, FALSE or NULL. Default = NULL, which will automatically set <code>classification = TRUE</code> if the number of unique response value ≤ 10 . Otherwise, it will be set as FALSE.
<code>...</code>	additional arguments.

Value

A list including the following items.

<code>model</code>	the model used in RaSE screening.
<code>criterion</code>	the criterion to choose the best subspace for each weak learner.
<code>B1</code>	the number of selected subspaces.
<code>B2</code>	the number of subspace candidates generated for each of B1 subspaces.
<code>n</code>	the sample size.
<code>p</code>	the dimension of data.
<code>D</code>	the maximal subspace size when generating random subspaces.
<code>iteration</code>	the number of iterations.
<code>selected.perc</code>	A list of length (<code>iteration+1</code>) recording the selected percentages of each feature in B1 subspaces. When it is of length 1, the result will be automatically transformed to a vector.
<code>scale</code>	a list of scaling parameters, including the scaling center and the scale parameter for each feature. Equals to NULL when the data is not scaled by RaScreen.

References

- Tian, Y. and Feng, Y., 2021. RaSE: A Variable Screening Framework via Random Subspace Ensembles.
- Tian, Y. and Feng, Y., 2021. RaSE: Random subspace ensemble classification. *Journal of Machine Learning Research*, 22, to appear.
- Chen, J. and Chen, Z., 2008. Extended Bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3), pp.759-771.
- Chen, J. and Chen, Z., 2012. Extended BIC for small-n-large-P sparse GLM. *Statistica Sinica*, pp.555-574.
- Schwarz, G., 1978. Estimating the dimension of a model. *The annals of statistics*, 6(2), pp.461-464.

See Also

[Rase](#), [RaRank](#).

Examples

```

set.seed(0, kind = "L'Ecuyer-CMRG")
train.data <- RaModel("screening", 1, n = 100, p = 100)
xtrain <- train.data$x
ytrain <- train.data$y

# test RaSE screening with linear regression model and BIC
fit <- RaScreen(xtrain, ytrain, B1 = 100, B2 = 50, iteration = 0, model = 'lm',
cores = 2, criterion = 'bic')

# Select D variables
RaRank(fit, selected.num = "D")

## Not run:
# test RaSE screening with knn model and 5-fold cross-validation MSE
fit <- RaScreen(xtrain, ytrain, B1 = 100, B2 = 50, iteration = 0, model = 'knn',
cores = 2, criterion = 'cv', cv = 5)

# Select n/logn variables
RaRank(fit, selected.num = "n/logn")

# test RaSE screening with SVM and 5-fold cross-validation MSE
fit <- RaScreen(xtrain, ytrain, B1 = 100, B2 = 50, iteration = 0, model = 'svm',
cores = 2, criterion = 'cv', cv = 5)

# Select n/logn variables
RaRank(fit, selected.num = "n/logn")

# test RaSE screening with logistic regression model and eBIC (gam = 0.5). Set iteration number = 1
train.data <- RaModel("screening", 6, n = 100, p = 100)
xtrain <- train.data$x
ytrain <- train.data$y

fit <- RaScreen(xtrain, ytrain, B1 = 100, B2 = 100, iteration = 1, model = 'logistic',
cores = 2, criterion = 'ebic', gam = 0.5)

# Select n/logn variables from the selected percentage after one iteration round
RaRank(fit, selected.num = "n/logn", iteration = 1)

## End(Not run)

```

Description

RaSE is a general ensemble classification framework to solve the sparse classification problem. In RaSE algorithm, for each of the B1 weak learners, B2 random subspaces are generated and the optimal one is chosen to train the model on the basis of some criterion.

Usage

```
Rase(
  xtrain,
  ytrain,
  xval = NULL,
  yval = NULL,
  B1 = 200,
  B2 = 500,
  D = NULL,
  dist = NULL,
  base = c("lda", "qda", "knn", "logistic", "tree", "svm", "randomforest", "gamma",
           "NULL"),
  criterion = NULL,
  ranking = TRUE,
  k = c(3, 5, 7, 9, 11),
  cores = 1,
  seed = NULL,
  iteration = 0,
  cutoff = TRUE,
  cv = 5,
  scale = FALSE,
  C0 = 0.1,
  kl.k = NULL,
  ...
)
```

Arguments

xtrain	n * p observation matrix. n observations, p features.
ytrain	n 0/1 observations.
xval	observation matrix for validation. Default = NULL. Useful only when criterion = 'validation'.
yval	0/1 observation for validation. Default = NULL. Useful only when criterion = 'validation'.
B1	the number of weak learners. Default = 200.
B2	the number of subspace candidates generated for each weak learner. Default = 500.
D	the maximal subspace size when generating random subspaces. Default = NULL, which is $\min(\sqrt{n_0}, \sqrt{n_1}, p)$ when base = 'qda' and is $\min(\sqrt{n}, p)$ otherwise.

dist	the distribution for features when generating random subspaces. Default = NULL, which represents the uniform distribution. First generate an integer d from $1, \dots, D$ uniformly, then uniformly generate a subset with cardinality d .
base	<p>the type of base classifier. Default = 'lda'.</p> <ul style="list-style-type: none"> • lda: linear discriminant analysis. lda in MASS package. • qda: quadratic discriminant analysis. qda in MASS package. • knn: k-nearest neighbor. knn, knn.cv in cLass package and knn3 in caret package. • logistic: logistic regression. glmnet in glmnet package. • tree: decision tree. rpart in rpart package. • svm: support vector machine. svm in e1071 package. • randomforest: random forest. randomForest in randomForest package. • gamma: Bayesian classifier for multivariate gamma distribution with independent marginals.
criterion	<p>the criterion to choose the best subspace for each weak learner. Default = 'ric' when base = 'lda', 'qda', 'gamma'; default = 'ebic' and set gam = 0 when base = 'logistic'; default = 'loo' when base = 'knn'; default = 'training' when base = 'tree', 'svm', 'randomforest'.</p> <ul style="list-style-type: none"> • ric: minimizing ratio information criterion with parametric estimation (Tian, Y. and Feng, Y., 2020). Available when base = 'lda', 'qda', 'gamma' or 'logistic'. • nric: minimizing ratio information criterion with non-parametric estimation (Tian, Y. and Feng, Y., 2020). Available when base = 'lda', 'qda', 'gamma' or 'logistic'. • training: minimizing training error. Not available when base = 'knn'. • loo: minimizing leave-one-out error. Only available when base = 'knn'. • validation: minimizing validation error based on the validation data. Available for all base classifiers. • cv: minimizing k-fold cross-validation error. k equals to the value of cv. Default = 5. Not available when base = 'gamma'. • aic: minimizing Akaike information criterion (Akaike, H., 1973). Available when base = 'lda' or 'logistic'. $AIC = -2 * \log\text{-likelihood} + S * 2$. • bic: minimizing Bayesian information criterion (Schwarz, G., 1978). Available when base = 'lda' or 'logistic'. $BIC = -2 * \log\text{-likelihood} + S * \log(n)$. • ebic: minimizing extended Bayesian information criterion (Chen, J. and Chen, Z., 2008; 2012). Need to assign value for gam. When gam = 0, it denotes the classical BIC. Available when base = 'lda' or 'logistic'. $EBIC = -2 * \log\text{-likelihood} + S * \log(n) + 2 * S * \text{gam} * \log(p)$.
ranking	whether the function outputs the selected percentage of each feature in B1 subspaces. Logistic, default = TRUE.
k	the number of nearest neighbors considered when base = 'knn'. Only useful when base = 'knn'. Default = (3, 5, 7, 9, 11).

cores	the number of cores used for parallel computing. Default = 1.
seed	the random seed assigned at the start of the algorithm, which can be a real number or NULL. Default = NULL, in which case no random seed will be set.
iteration	the number of iterations. Default = 0.
cutoff	whether to use the empirically optimal threshold. Logistic, default = TRUE. If it is FALSE, the threshold will be set as 0.5.
cv	the number of cross-validations used. Default = 5. Only useful when <code>criterion = 'cv'</code> .
scale	whether to normalize the data. Logistic, default = FALSE.
C0	a positive constant used when <code>iteration > 1</code> . See Tian, Y. and Feng, Y., 2020 for details.
k1.k	the number of nearest neighbors used to estimate RIC in a non-parametric way. Default = NULL, which means that $k_0 = \text{floor}(\sqrt{n_0})$ and $k_1 = \text{floor}(\sqrt{n_1})$. See Tian, Y. and Feng, Y., 2020 for details. Only available when <code>criterion = 'nric'</code> .
...	additional arguments.

Value

An object with S3 class 'RaSE'.

marginal	the marginal probability for each class.
base	the type of base classifier.
criterion	the criterion to choose the best subspace for each weak learner.
B1	the number of weak learners.
B2	the number of subspace candidates generated for each weak learner.
D	the maximal subspace size when generating random subspaces.
iteration	the number of iterations.
fit.list	sequence of B1 fitted base classifiers.
cutoff	the empirically optimal threshold.
subspace	sequence of subspaces corresponding to B1 weak learners.
ranking	the selected percentage of each feature in B1 subspaces.
scale	a list of scaling parameters, including the scaling center and the scale parameter for each feature. Equals to NULL when the data is not scaled in RaSE model fitting.

References

- Tian, Y. and Feng, Y., 2021. RaSE: Random subspace ensemble classification. *Journal of Machine Learning Research*, 22, to appear.
- Chen, J. and Chen, Z., 2008. Extended Bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3), pp.759-771.

Chen, J. and Chen, Z., 2012. Extended BIC for small-n-large-P sparse GLM. *Statistica Sinica*, pp.555-574.

Akaike, H., 1973. Information theory and an extension of the maximum likelihood principle. In *2nd International Symposium on Information Theory, 1973* (pp. 267-281). Akademiai Kiado.

Schwarz, G., 1978. Estimating the dimension of a model. *The annals of statistics*, 6(2), pp.461-464.

See Also

[predict.RaSE](#), [RaModel](#), [print.RaSE](#), [RaPlot](#), [RaScreen](#).

Examples

```
set.seed(0, kind = "L'Ecuyer-CMRG")
train.data <- RaModel("classification", 1, n = 100, p = 50)
test.data <- RaModel("classification", 1, n = 100, p = 50)
xtrain <- train.data$x
ytrain <- train.data$y
xtest <- test.data$x
ytest <- test.data$y

# test RaSE classifier with LDA base classifier
fit <- Rase(xtrain, ytrain, B1 = 100, B2 = 50, iteration = 0, base = 'lda',
cores = 2, criterion = 'ric')
mean(predict(fit, xtest) != ytest)

## Not run:
# test RaSE classifier with LDA base classifier and 1 iteration round
fit <- Rase(xtrain, ytrain, B1 = 100, B2 = 50, iteration = 1, base = 'lda',
cores = 2, criterion = 'ric')
mean(predict(fit, xtest) != ytest)

# test RaSE classifier with QDA base classifier and 1 iteration round
fit <- Rase(xtrain, ytrain, B1 = 100, B2 = 50, iteration = 1, base = 'qda',
cores = 2, criterion = 'ric')
mean(predict(fit, xtest) != ytest)

# test RaSE classifier with knn base classifier
fit <- Rase(xtrain, ytrain, B1 = 100, B2 = 50, iteration = 0, base = 'knn',
cores = 2, criterion = 'loo')
mean(predict(fit, xtest) != ytest)

# test RaSE classifier with logistic regression base classifier
fit <- Rase(xtrain, ytrain, B1 = 100, B2 = 50, iteration = 0, base = 'logistic',
cores = 2, criterion = 'ebic', gam = 0)
mean(predict(fit, xtest) != ytest)

# test RaSE classifier with svm base classifier
fit <- Rase(xtrain, ytrain, B1 = 100, B2 = 50, iteration = 0, base = 'svm',
cores = 2, criterion = 'training')
mean(predict(fit, xtest) != ytest)

# test RaSE classifier with random forest base classifier
```

```
fit <- Rase(xtrain, ytrain, B1 = 20, B2 = 10, iteration = 0, base = 'randomforest',  
cores = 2, criterion = 'cv', cv = 3)  
mean(predict(fit, xtest) != ytest)  
  
## End(Not run)
```


Index

`glmnet`, [9](#), [13](#)

`knn`, [9](#), [13](#)

`knn.cv`, [9](#), [13](#)

`knn3`, [9](#), [13](#)

`knnreg`, [9](#)

`lda`, [8](#), [13](#)

`predict.RaSE`, [2](#), [15](#)

`print.RaSE`, [3](#), [15](#)

`qda`, [9](#), [13](#)

`RaModel`, [4](#), [15](#)

`randomForest`, [9](#), [13](#)

`ranger`, [9](#)

`RaPlot`, [5](#), [15](#)

`RaRank`, [6](#), [11](#)

`RaScreen`, [5](#), [7](#), [15](#)

`Rase`, [2](#), [3](#), [5](#), [6](#), [11](#), [11](#)

`rpart`, [9](#), [13](#)

`svm`, [9](#), [13](#)