Package 'cvLM'

August 1, 2024

Type Package

Title Cross-Validation for Linear & Ridge Regression Models

Version 1.0.4

Date 2024-07-30

Description Efficient implementations of cross-validation techniques for linear and ridge regression models, leveraging 'C++' code with 'Rcpp', 'RcppParallel', and 'Eigen' libraries. It supports leave-one-out, generalized, and K-fold cross-validation methods, utilizing 'Eigen' matrices for high performance. Methodology references: Hastie, Tibshirani, and Friedman (2009) <doi:10.1007/978-0-387-84858-7>.

License MIT + file LICENSE

Imports stats, Rcpp (>= 1.0.13), RcppParallel (>= 5.1.8)

LinkingTo Rcpp, RcppParallel, RcppEigen

SystemRequirements GNU make

NeedsCompilation yes

Author Philip Nye [aut, cre]

Maintainer Philip Nye <`phipnye@proton.me`>

Repository CRAN

Date/Publication 2024-08-01 09:30:09 UTC

Contents

cvLM-package	*Cross-validation for linear and ridge regression models*

Description

This package provides efficient implementations of cross-validation techniques for linear and ridge regression models, leveraging C++ code with Rcpp, RcppParallel, and Eigen libraries. It supports leave-one-out, generalized, and K-fold cross-validation methods, utilizing Eigen matrices for high performance.

Usage

```
cvLM(object, ...)
## S3 method for class 'formula'
cvLM(object, data, subset, na.action, K.vals = 10L, lambda = 0,
     generalized = FALSE, seed = 1L, n.threads = 1L, ...)
## S3 method for class 'lm'
cvLM(object, data, K.vals = 10L, lambda = 0,
     generalized = FALSE, seed = 1L, n.threads = 1L, ...)
## S3 method for class 'glm'
cvLM(object, data, K.vals = 10L, lambda = 0,
     generalized = FALSE, seed = 1L, n.threads = 1L, ...)
```

Arguments

object	a formula, a linear model (lm), or a generalized linear model (glm) object.
data	a data.frame containing the variables in the model.
subset	an optional vector specifying a subset of observations to be used in the fitting process. See model.frame for more details.
na.action	a function that indicates how to handle NA values in the data. See model.frame for more details.
K.vals	an integer vector specifying the number of folds for cross-validation.
lambda	a non-negative numeric scalar specifying the regularization parameter for ridge regression.
generalized	a logical value indicating whether to compute generalized or ordinary cross-validation. Defaults to FALSE for ordinary cross-validation.
seed	a single integer value specifying the seed for random number generation.
n.threads	a single positive integer value specifying the number of threads for parallel computation.
...	additional arguments. Currently, these do not affect the function's behavior.

Details

The `cvLM` function is a generic function that dispatches to specific methods based on the class of the `object` argument.

The `cvLM.formula` method performs cross-validation for linear and ridge regression models specified using a formula interface.

The `cvLM.lm` method performs cross-validation for linear regression models.

The `cvLM.glm` method performs cross-validation for generalized linear models. It currently supports only gaussian family with identity link function.

The cross-validation process involves splitting the data into `K` folds, fitting the model on `K-1` folds, and evaluating its performance on the remaining fold. This process is repeated `K` times, each time with a different fold held out for testing.

The `cvLM` functions use closed-form solutions for leave-one-out and generalized cross-validation and efficiently handle the K-fold cross-validation process, optionally using multithreading for faster computation when working with larger data.

Value

A `data.frame` consisting of columns representing the number of folds, the cross-validation result, and the seed used for the computation.

Author(s)

Philip Nye, `<phipnye@proton.me>`

References

Bates D, Eddelbuettel D (2013). "Fast and Elegant Numerical Linear Algebra Using the RcppEigen Package." Journal of Statistical Software, 52(5), 1-24. doi:10.18637/jss.v052.i05.

Aggarwal, C. C. (2020). Linear Algebra and Optimization for Machine Learning: A Textbook. Springer Cham. doi:10.1007/9783030403447.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.). Springer New York, NY. doi:10.1007/978038784858-7.

See Also

`formula`, `lm`, `glm`

Examples

```
data(mtcars)
n <- nrow(mtcars)

# Formula method
cvLM(
  mpg ~ .,
  data = mtcars,
  K.vals = n,    # Leave-one-out CV
  lambda = 10    # Shrinkage parameter of 10
)

# lm method
my.lm <- lm(mpg ~ ., data = mtcars)
cvLM(
  my.lm,
  data = mtcars,
  K.vals = c(5L, 8L), # Perform both 5- and 8-fold CV
  n.threads = 8L,     # Allow up to 8 threads for computation
  seed = 1234L
)

# glm method
my.glm <- glm(mpg ~ ., data = mtcars)
cvLM(
  my.glm,
  data = mtcars,
  K.vals = n, generalized = TRUE # Use generalized CV
)
```

`grid.search` *Grid Search for Optimal Lambda in Ridge Regression*

Description

Performs a grid search to find the optimal lambda value for ridge regression.

Usage

```
grid.search(formula, data, subset, na.action, K = 10L,
            generalized = FALSE, seed = 1L, n.threads = 1L,
            max.lambda = 10000, precision = 0.1, verbose = TRUE)
```

Arguments

`formula`	a `formula` specifying the model.
`data`	a `data.frame` containing the variables in the model.
`subset`	an optional vector specifying a subset of observations to be used in the fitting process. See `model.frame` for more details.
`na.action`	a function that indicates what should happen when the data contain `NA`s. See `model.frame` for more details.
`K`	an integer specifying the number of folds for cross-validation.
`generalized`	a logical indicating whether to compute generalized or ordinary cross-validation.
`seed`	an integer specifying the seed for random number generation.
`n.threads`	an integer specifying the number of threads for parallel computation.

`max.lambda`	a numeric specifying the maximum value of lambda to search.
`precision`	a numeric specifying the precision of the lambda search.
`verbose`	a logical indicating whether to display a progress bar during the computation process.

Details

This function performs a grid search to determine the optimal regularization parameter lambda for ridge regression. It evaluates lambda values starting from 0, incrementing by the specified precision up to the maximum lambda provided. The function utilizes cross-validation to assess the performance of each lambda value and selects the one that minimizes the cross-validation error.

Value

A `list` with the following components:

CV the minimum cross-validation error.

lambda the corresponding optimal lambda value.

Examples

```
data(mtcars)
grid.search(
  formula = mpg ~ .,
  data = mtcars,
  K = 5L,           # Use 5-fold CV
  max.lambda = 100, # Search values between 0 and 100
  precision = 0.01, # Increment in steps of 0.01
  verbose = FALSE,  # Disable progress bar
)


set.seed(1L)
n <- 50002L

DF <- data.frame(
  x1 = rnorm(n),
  x2 = runif(n),
  x3 = rlogis(n),
  x4 = rnorm(n),
  x5 = rcauchy(n),
  x6 = rexp(n)
)

DF <- transform(
  DF,
  y = 3.1 * x1 + 1.8 * x2 + 0.9 * x3 + 1.2 * x4 + x5 + 0.6 * x6 + rnorm(n, sd = 10)
)

grid.search(
  formula = y ~ .,
  data = DF,
  K = 10L,          # Use 10-fold CV
  max.lambda = 100, # Search values between 0 and 100
  precision = 1,    # Increment in steps of 1
  n.threads = 10L   # Use 10 threads
)

grid.search(
  formula = y ~ .,
  data = DF,
  K = n, generalized = TRUE, # Use generalized CV
  max.lambda = 10000,        # Search values between 0 and 10000
  precision = 10,            # Increment in steps of 10
)
```

reg.table *Create Regression Tables in LaTeX or HTML*

Description

reg.table generates formatted regression tables in either LaTeX or HTML format for one or more linear regression models. The tables include coefficient estimates, standard errors, and various model statistics.

Usage

```
reg.table(models, type = c("latex", "html"), split.size = 4L, n.digits = 3L,
          big.mark = "", caption = "Regression Results", spacing = 5L,
          stats = c("AIC", "BIC", "CV", "r.squared", "adj.r.squared",
                    "fstatistic", "nobs"),
          ...)
```

Arguments

models	A list of linear regression models (objects of class lm or glm with gaussian family and identity link function).
type	A string specifying the output format. Must be one of "latex" or "html".
split.size	An integer specifying the number of models per table. Tables with more models than this number will be split.
n.digits	An integer specifying the number of decimal places for numerical values.
big.mark	A string to use for thousands separators in numeric formatting.
caption	A string for the table caption.
spacing	A numeric specifying the spacing between columns in the output table.
stats	A character vector specifying which model statistics to include in the table. Options include:

	`AIC` Akaike Information Criterion
	`BIC` Bayesian Information Criterion
	`CV` Cross-validation score
	`r.squared` R-squared
	`adj.r.squared` Adjusted R-squared
	`fstatistic` F-statistic
	`nobs` Number of observations
`...`	Additional arguments passed to `cvLM`.

Details

The function generates tables summarizing linear regression models, either in LaTeX or HTML format. The tables include coefficients, standard errors, and optional statistics such as AIC, BIC, cross-validation scores, and more. The output is formatted based on the specified type and options. The names of the models list will be used as column headers if provided.

Value

A character vector of length equal to the number of tables with each element containing the LaTeX or HTML code for a regression table.

Examples

```
data(mtcars)

# Create a list of 6 different linear regression models with names
# Names get used for column names of the table
models <- list(
  `Model 1` = lm(mpg ~ wt + hp, data = mtcars),
  `Model 2` = lm(mpg ~ wt + qsec, data = mtcars),
  `Model 3` = lm(mpg ~ wt + hp + qsec, data = mtcars),
  `Model 4` = lm(mpg ~ wt + cyl, data = mtcars),
  `Model 5` = lm(mpg ~ wt + hp + cyl, data = mtcars),
  `Model 6` = lm(mpg ~ hp + qsec + drat, data = mtcars)
)

# Example usage for LaTeX
reg.table(
  models,
  "latex",
  split.size = 3L,
  n.digits = 3L,
  big.mark = ",",
  caption = "My regression results",
  spacing = 7.5,
  stats = c("AIC", "BIC", "CV"),
  data = mtcars,
  K.vals = 5L,
  seed = 123L
)
```

```
# Example usage for HTML
reg.table(
  models,
  "html",
  split.size = 3L,
  n.digits = 3L,
  big.mark = ",",
  caption = "My regression results",
  spacing = 7.5,
  stats = c("AIC", "BIC", "CV"),
  data = mtcars,
  K.vals = 5L,
  seed = 123L
)
```

Index