Package ‘rollRegres’

November 25, 2019

Type Package
Title Fast Rolling and Expanding Window Linear Regression
Version 0.1.3
Description Methods for fast rolling and expanding linear regression models. That is, series of linear regression models estimated on either an expanding window of data or a moving window of data. The methods use rank-one updates and downdates of the upper triangular matrix from a QR decomposition (see Dongarra, Moler, Bunch, and Stewart (1979) <doi:10.1137/1.9781611971811>).

Copyright (c) 2018-2019 Benjamin Christoffersen, except dchud.f and dchdd.f. They are originally from LINPACK and are in the public domain in the United States. They are modified by Madeleine Thompson.

License GPL-2
Encoding UTF-8
LazyData true
LinkingTo Rcpp, RcppArmadillo
Imports Rcpp, checkmate
Suggests knitr, rmarkdown, testthat, zoo, roll, microbenchmark, RcppParallel
VignetteBuilder knitr
RoxygenNote 7.0.0
BugReports https://github.com/boennecd/rollRegres/issues
SystemRequirements C++11
URL https://github.com/boennecd/rollRegres
NeedsCompilation yes
Author Benjamin Christoffersen [cre, aut], Madeleine Thompson [cph]
Maintainer Benjamin Christoffersen <boennecd@gmail.com>
Repository CRAN
Date/Publication 2019-11-25 09:20:02 UTC
**R topics documented:**

- roll_regres .......................................................... 2
- roll_regres.fit ....................................................... 3

---

**Index**

| roll_regres | Fitting Rolling and Expanding Linear Models |

---

**Description**

Method for fast rolling and expanding regression models. I.e., linear models estimated over a moving window or expanding window of data. The function assumes that data is ordered.

**Usage**

```r
roll_regres(formula, data, width, contrasts = NULL,
             do_compute = character(), grp = NULL, do_downdates = TRUE,
             min_obs = NULL)
```

**Arguments**

- `formula` as formula in `lm`.
- `data` an optional `data.frame` containing the variables in the model.
- `width` integer with the width of the moving window. Only used if `do_downdates == TRUE`.
- `contrasts` list passed to `model.matrix`'s `contrasts.arg` argument.
- `do_compute` character vector with elements "sigmas", "r.squareds", and/or "1_step_forecasts" for additional output to be computed. See "Details" in `roll_regres`.
- `grp` integer vector to be used if you e.g., want to run the regression over weekly blocks of data. See "Details" in `roll_regres`.
- `do_downdates` logical which is TRUE if you want a rolling window regressions. Otherwise, an expanding window is used.
- `min_obs` positive integer with minimum number of observation that are required in a window. Useful if there are gaps in `grp` or unequal number of observations for each `grp`.

**Details**

do_compute can contain "sigmas" if you want the estimated standard deviation of the residuals, "r.squareds" for the $R^2$ of the models, and "1_step_forecasts" for the out-of-sample forecast for the next periods value.

 grp is a sorted integer vector if you want to make "block" updates. E.g., `grp` could be an integer vector with the week number. The width argument is relative to the `grp` argument if the `grp` argument is not NULL. The indices of `grp` should match with the other data objects.

See vignette("Comparisons", package = "rollRegres") for further examples.
Value

List with vector and matrices with the computed output. See the do_compute argument.

See Also

roll_regres.fit for method that avoids the call to e.g., model.frame.

Examples

# simulate data
set.seed(29132867)
n <- 50
p <- 2
X <- cbind(1, matrix(rnorm(p * n), ncol = p))
y <- drop(X %*% c(1, -1, 1)) + rnorm(n)
df <- data.frame(y, X[, -1])

# compute coefs
out <- roll_regres(y ~ X1 + X2, df, width = 45L)
tail(out$coefs)

# compute more output
out <- roll_regres(
  y ~ X1 + X2, df, width = 45L,
  do_compute = c("sigmas", "r.squareds", "1_step_forecasts")
)
lapply(out, tail)

roll_regres.fit  Fitter Function for Rolling and Expanding Linear Models

Description

Function with a few validations before calling C++ code.

Usage

roll_regres.fit(x, y, width, do_compute = character(), grp = NULL,
  do_downdates = TRUE, min_obs = NULL)

Arguments

x  
design matrix of dimension n * p.

y  
numeric vector of observations of length n.

width  
integer with the width of the moving window. Only used if do_downdates == TRUE.

do_compute  
character vector with elements "sigmas", "r.squareds", and/or "1_step_forecasts" for additional output to be computed. See "Details" in roll_regres.
**roll_regres.fit**

- **grp**: integer vector to be used if you e.g., want to run the regression over weekly blocks of data. See "Details" in `roll_regres`.

- **do_downdates**: logical which is TRUE if you want a rolling window regressions. Otherwise, an expanding window is used.

- **min_obs**: positive integer with minimum number of observation that are required in a window. Useful if there are gaps in `grp` or unequal number of observations for each `grp`.

**Details**

First, the `dqrdc` routine from LINPACK is used to form the QR decomposition for the first window of data using Householder transformations without pivoting. Then, the LINPACK `dchud` and `dchdd` routines are used to update and downdate the Cholesky decomposition (the R matrix in the QR decomposition).

Notice that unlike `lm`, there are no checks of the rank of the matrix.

**Value**

Same as `roll_regres`.

**References**


**See Also**

`roll_regres` for method similar to `lm`.

**Examples**

```r
# simulate data
set.seed(9623556)
n <- 50
p <- 2
X <- cbind(1, matrix(rnorm(p * n), ncol = p))
y <- drop(X %*% c(1, -1, 1)) + rnorm(n)

# compute coefs
out <- roll_regres.fit(x = X, y = y, width = 45L)
tail(out$coefs)
```
Index

lm, 2, 4
model.frame, 3
model.matrix.default, 2
roll_regres, 2, 2, 3, 4
roll_regres.fit, 3, 3