

Package ‘clr’

August 31, 2026

Type Package

Title Curve Linear Regression via Dimension Reduction

Version 0.1.3

Maintainer Amandine Pierrot <amandine.m.pierrot@gmail.com>

Copyright EDF R&D 2017

Description A new methodology for linear regression with both curve response and curve regressors, which is described in Cho, Goude, Brossat and Yao (2013) <[doi:10.1080/01621459.2012.722900](https://doi.org/10.1080/01621459.2012.722900)> and (2015) <[doi:10.1007/978-3-319-18732-7_3](https://doi.org/10.1007/978-3-319-18732-7_3)>. The key idea behind this methodology is dimension reduction based on a singular value decomposition in a Hilbert space, which reduces the curve regression problem to several scalar linear regression problems.

License LGPL (>= 2.0)

Encoding UTF-8

LazyData true

Depends R (>= 2.10)

Imports magrittr, lubridate, dplyr, stats

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Amandine Pierrot [cre, aut],
Qiwei Yao [ctb],
Haeran Cho [ctb],
Yannig Goude [ctb],
Tony Aldon [ctb],
EDF [cph, fnd]

Repository CRAN

Date/Publication 2026-08-31 20:50:03 UTC

Contents

clr-package	2
-----------------------	---

clr	3
clrdata	5
clust_test	6
clust_train	6
gb_load	7
predict.clr	8
Index	10

clr-package	<i>Curve Linear Regression</i>
-------------	--------------------------------

Description

clr provides functions for curve linear regression via dimension reduction.

Details

The package implements a new methodology for linear regression with both curve response and curve regressors, which is described in Cho et al. (2013) and Cho et al. (2015). The CLR model performs a data-driven dimension reduction, based on a singular value decomposition in a Hilbert Space, as well as a data transformation so that the relationship between the transformed data is linear and can be captured by simple regression models.

Author(s)

Amandine Pierrot <amandine.m.pierrot@gmail.com>
with contributions and help from Qiwei Yao, Haeran Cho, Yannig Goude and Tony Aldon.

References

These provide details for the underlying clr methods.

Cho, H., Y. Goude, X. Brossat, and Q. Yao (2013) Modelling and Forecasting Daily Electricity Load Curves: A Hybrid Approach. *Journal of the American Statistical Association* 108: 7-21.

Cho, H., Y. Goude, X. Brossat, and Q. Yao (2015) Modelling and Forecasting Daily Electricity Load via Curve Linear Regression. In *Modeling and Stochastic Learning for Forecasting in High Dimension*, edited by Anestis Antoniadis and Xavier Brossat, 35-54, Springer.

clr

*Curve Linear Regression via dimension reduction***Description**

Fits a curve linear regression (CLR) model to data, using dimension reduction based on singular value decomposition.

Usage

```
clr(
  Y,
  X,
  clust = NULL,
  qx_estimation = list(method = "pctvar", param = 0.999),
  ortho_Y = TRUE,
  qy_estimation = list(method = "pctvar", param = 0.999),
  d_estimation = list(method = "cor", param = 0.5)
)
```

Arguments

Y	An object of class <code>clrdata</code> or <code>matrix</code> , of the response curves (one curve a row).
X	An object of class <code>clrdata</code> or <code>matrix</code> , of the regressor curves (one curve a row).
clust	If needed, a list of row indices for each cluster, to obtain (approximately) homogeneous dependence structure inside each cluster.
qx_estimation	A list containing both values for 'method' (among 'ratio', 'ratioM', 'pctvar', 'fixed') and for 'param' (depending on the selected method), in order to choose how to estimate the dimension of X (in the sense that its Karhunen-Loève decomposition has qx terms only).
ortho_Y	If TRUE then Y is orthogonalized.
qy_estimation	Same as for qx_estimation, if ortho_Y is set to TRUE.
d_estimation	A list containing both values for 'method' (among 'ratio', 'pctvar', 'cor') and for 'param' (depending on the selected method), in order to choose how to estimate the correlation dimension.

Value

An object of class `clr`, which can be used to compute predictions. This `clr` object is a list of lists: one list by cluster of data, each list including:

residuals	The matrix of the residuals of <code>d_hat</code> simple linear regressions.
b_hat	The vector of the estimated coefficient of the <code>d_hat</code> simple straight line regressions.
eta	The matrix of the projections of X.

<code>xi</code>	The matrix of the projections of Y.
<code>qx_hat</code>	The estimated dimension of X.
<code>qy_hat</code>	The estimated dimension of Y.
<code>d_hat</code>	The estimated correlation dimension.
<code>X_mean</code>	The mean of the regressor curves.
<code>X_sd</code>	The standard deviation of the regressor curves.
<code>Y_mean</code>	The mean of the response curves.
<code>ortho_Y</code>	The value which was selected for <code>ortho_Y</code> .
<code>GAMMA</code>	The standardized transformation for X.
<code>INV_DELTA</code>	The standardized transformation for Y to predict if <code>ortho_Y</code> was set to TRUE.
<code>phi</code>	The eigenvectors for Y to predict if <code>ortho_Y</code> was set to FALSE.
<code>idx</code>	The indices of the rows selected from X and Y for the current cluster.

See Also

[clr-package](#), [clrdata](#) and [predict.clr](#).

Examples

```
library(clr)
data(gb_load)
data(clust_train)

clr_load <- clrdata(x = gb_load$ENGLAND_WALES_DEMAND,
                   order_by = gb_load$TIMESTAMP,
                   support_grid = 1:48)

## data cleaning: replace zeros with NA
clr_load[rowSums((clr_load == 0) * 1) > 0, ] <- NA
matplot(t(clr_load), ylab = 'Daily loads', type = 'l')

Y <- clr_load[2:nrow(clr_load), ]
X <- clr_load[1:(nrow(clr_load) - 1), ]

begin_pred <- which(substr(rownames(Y), 1, 4) == '2016')[1]
Y_train <- Y[1:(begin_pred - 1), ]
X_train <- X[1:(begin_pred - 1), ]

## Example without any cluster
model <- clr(Y = Y_train, X = X_train)

## Example with clusters
model <- clr(Y = Y_train, X = X_train, clust = clust_train)
```

clrdata	Create an object of clrdata
---------	-----------------------------

Description

clrdata is used to create a clrdata object from raw data inputs.

Usage

```
clrdata(x, order_by, support_grid)
```

Arguments

x	A vector containing the time series values
order_by	A corresponding vector of unique time-dates - must be of class 'POSIXct'
support_grid	A vector corresponding to the support grid of functional data

Value

An object of class clrdata with one function a row. As it inherits the matrix class, all matrix methods remain valid. If time-dates are missing in x, corresponding NA functions are added by clrdata so that time sequence is preserved between successive rows.

Examples

```
library(clr)
data(gb_load)

clr_load <- clrdata(x = gb_load$ENGLAND_WALES_DEMAND,
                   order_by = gb_load$TIMESTAMP,
                   support_grid = 1:48)

head(clr_load)
dim(clr_load)
summary(clr_load)

matplot(t(clr_load), ylab = 'Daily loads', type = 'l')
lines(colMeans(clr_load, na.rm = TRUE),
      col = 'black', lwd = 2)

clr_weather <- clrdata(x = gb_load$TEMPERATURE,
                      order_by = gb_load$TIMESTAMP,
                      support_grid = 1:48)

summary(clr_weather)
plot(1:48,
     colMeans(clr_weather, na.rm = TRUE),
     xlab = 'Instant', ylab = 'Mean of temperatures',
     type = 'l', col = 'cornflowerblue')
```

`clust_test`*Electricity load example: clusters on test set*

Description

A list with observations by cluster for prediction

Usage`clust_test`**Format**

A list of length 14:

14 clusters of loads, depending on both daily and seasonal classification, banking holidays being removed

Author(s)

Amandine Pierrot <amandine.m.pierrot@gmail.com>

`clust_train`*Electricity load example: clusters on train set*

Description

A list with observations by cluster for fitting

Usage`clust_train`**Format**

A list of length 14:

14 clusters of loads, depending on both daily and seasonal classification, banking holidays being removed

Author(s)

Amandine Pierrot <amandine.m.pierrot@gmail.com>

gb_load*Electricity load from Great Britain*

Description

A dataset containing half-hourly electricity load from Great Britain from 2011 to 2016, together with observed temperatures. Temperatures are computed from weather stations all over the country. It is a weighted averaged temperature depending on population geographical distribution.

Usage

gb_load

Format

A data frame with 105216 rows and 7 variables:

SETTLEMENT_DATE date, the time zone being Europe/London

SETTLEMENT_PERIOD time of the day

TIMESTAMP date-time, the time zone being Europe/London

ENGLAND_WALES_DEMAND British electric load, measured in MW, on average over the half hour

TEMPERATURE observed temperature in Celsius

MV percentage of missing values when averaging over weather stations, depending on the weight of the station

DAY_TYPE type of the day of the week, from 1 for Sunday to 7 for Saturday, 8 being banking holidays

Author(s)

Amandine Pierrot <amandine.m.pierrot@gmail.com>

Source

National Grid

National Centers for Environmental Information

predict.clr

*Prediction from fitted CLR model(s)***Description**

Takes a fitted clr object produced by `clr()` and produces predictions given a new set of functions or the original values used for the model fit.

Usage

```
## S3 method for class 'clr'
predict(
  object,
  newX = NULL,
  newclust = NULL,
  newXmean = NULL,
  simplify = FALSE,
  ...
)
```

Arguments

<code>object</code>	A fitted clr object produced by <code>clr()</code> .
<code>newX</code>	An object of class <code>clrdata</code> or a matrix with one function a row. If this is not provided then predictions corresponding to the original data are returned. If <code>newX</code> is provided then it should contain the same type of functions as the original ones (same dimension, same clusters eventually, ...).
<code>newclust</code>	A new list of indices to obtain (approximately) homogeneous dependence structure inside each cluster of functions.
<code>newXmean</code>	To complete when done
<code>simplify</code>	If TRUE, one matrix of predicted functions is returned instead of a list of matrices (one matrix by cluster). In the final matrix, rows are sorted by increasing row numbers.
<code>...</code>	Further arguments are ignored.

Value

predicted functions

Examples

```
library(clr)
data(gb_load)

clr_load <- clrdata(x = gb_load$ENGLAND_WALES_DEMAND,
  order_by = gb_load$TIMESTAMP,
```

```

support_grid = 1:48)

# data cleaning: replace zeros with NA
clr_load[rowSums((clr_load == 0) * 1) > 0, ] <- NA

Y <- clr_load[2:nrow(clr_load), ]
X <- clr_load[1:(nrow(clr_load) - 1), ]

begin_pred <- which(substr(rownames(Y), 1, 4) == '2016')[1]
Y_train <- Y[1:(begin_pred - 1), ]
X_train <- X[1:(begin_pred - 1), ]
Y_test <- Y[begin_pred:nrow(Y), ]
X_test <- X[begin_pred:nrow(X), ]

## Example without any cluster
model <- clr(Y = Y_train, X = X_train)

pred_on_train <- predict(model)
head(pred_on_train[[1]])

pred_on_test <- predict(model, newX = X_test)
head(pred_on_test[[1]])

## Example with clusters
model <- clr(Y = Y_train, X = X_train, clust = clust_train)

pred_on_train <- predict(model)
str(pred_on_train)
head(pred_on_train[[1]])

pred_on_test <- predict(model, newX = X_test, newclust = clust_test,
                        simplify = TRUE)
str(pred_on_test)
head(pred_on_test)

# With dates as row names
rownames(pred_on_test) <- rownames(Y_test)[as.numeric(rownames(pred_on_test))]
```

Index

- * **datasets**

- clust_test, [6](#)
 - clust_train, [6](#)
 - gb_load, [7](#)

- * **package**

- clr-package, [2](#)

clr, [3](#)

clr-package, [2](#)

clrdata, [4](#), [5](#)

clust_test, [6](#)

clust_train, [6](#)

gb_load, [7](#)

predict.clr, [4](#), [8](#)