Package ‘dlim’

February 29, 2024

Type Package
Date 2024-02-25
Title Distributed Lag Interaction Model
Version 0.1.0
Description Collection of functions for fitting and interpreting distributed lag interaction models (DLIM). A DLIM regresses a scalar outcome on repeated measures of exposure and allows for modification by a continuous variable. Includes a dlim() function for fitting, predict() function for inference, and plotting functions for visualization. Details on methodology are described in Demateis et al. (2024) <doi:10.1002/env.2843>.
License GPL-3
Encoding UTF-8
LazyData true
RoxygenNote 7.2.3
Imports dlnm, ggplot2, mgcv, reshape2, rlang, splines, tsModel, viridis
Depends R (>= 2.10)
Suggests rmarkdown, knitr
VignetteBuilder knitr
NeedsCompilation no
Author Danielle Demateis [aut, cre] (<https://orcid.org/0009-0003-0785-3962>), Kayleigh Keller [aut] (<https://orcid.org/0000-0002-9423-2704>), Ander Wilson [aut] (<https://orcid.org/0000-0003-4774-3883>)
Maintainer Danielle Demateis <Danielle.Demateis@colostate.edu>
Repository CRAN
Date/Publication 2024-02-29 20:42:42 UTC

R topics documented:

dlim-package ................................................................. 2
cross_basis ................................................................. 3
The package `dlim` contains functions to fit, perform inference and estimation on, and visualize a distributed lag interaction model (DLIM).

Modelling framework

A distributed lag interaction model (DLIM) is an extension of a distributed lag model. A DLIM regresses an outcome onto repeated measures of an exposure and allows for associations to vary by a single continuous modifier. More details on methodology are provided in the reference listed below.

Functions and data included in the package

To fit a DLIM using this package, use the `dlim` function, which calls the `cross_basis` function to create the cross-basis and estimates regression coefficients using `gam` from `mgcv` package.

The `predict.dlim` S3 function provides point-wise or cumulative effect estimates and uncertainty measures.

The `plot_DLF` and `plot_cumulative` functions provide plots of the modified distributed lag functions and the cumulative effect estimate curve.

Additional information

Additional details on the package `dlim` are available in the vignette, available by typing: `vignette("dlimOverview")`

The `dlim` package is available on the Comprehensive R Archive Network (CRAN). A development website is available on GitHub (`github.com/ddemateis/dlim`). Please use `citation("dlim")` to cite this package.
cross_basis

Author(s)
Danielle Demateis, Kayleigh Keller, and Ander Wilson
Maintainer: Danielle Demateis <Danielle.Demateis@colostate.edu>

References

See Also
Type 'vignette(dlimOverview)' for a detailed description.

---

cross_basis Build crossbasis

Description
Creates cross-basis using natural splines for regression in DLIM

Usage
cross_basis(
x, 
M, 
L = NULL, 
argmod = list(), 
arglag = list(), 
model_type = "standard"
)

Arguments
x a numeric time series vector of length n or matrix of lagged exposures (columns) for n individuals (rows)
M a numeric vector of length n containing modifier values
L a numeric vector of length 1 containing the number of lag terms. This is required if x is vector, and is not used if x is a matrix.
argmod a list: $fun is the spline function for the modifier ("ps" or "cr" to penalize), $arg is a list of arguments for the spline function (must be named by argument), $df is the degrees of freedom, $sp is optional smoothing parameter
arglag a list: $fun is the spline function for the lag ("ps" or "cr" to penalize), $arg is a list of arguments for the spline function (must be named by argument), $df is the degrees of freedom, $sp is optional smoothing parameter
model_type "linear" for a DLIM with linear interaction, "quadratic" for a DLIM with quadratic interaction, "standard" for a DLIM with splines
Value

This function returns a list of 5 or 6 elements:

- \( cb \) cross-basis (matrix)
- \( B_{\text{lag}} \) lag basis (basis matrix)
- \( B_{\text{mod}} \) modifier basis (basis matrix)
- \( df_{\text{l}} \) lag degrees of freedom (numeric)
- \( df_{\text{m}} \) modifier degrees of freedom (numeric)
- \( L \) number of lags (numeric)
- \( S_{\text{list}} \) lag and modifier penalty matrices, if penalizing (list)

See Also

dlim

Type vignette('dlimOverview') for a detailed description.

dlim

Fit DLIM

Description

Fit distributed lag interaction model

Usage

dlim(
y, x, modifiers, z = NULL, df_m, df_l, penalize = TRUE, pen_fn = "ps", mod_args = NULL, lag_args = NULL, fit_fn = "gam", model_type = "standard", ID = NULL, ...
)

Arguments

- **y**: vector of response values (class "numeric")
- **x**: matrix of exposure history (columns) for individuals (rows) (class "matrix")
- **modifiers**: vector of modifying values (class "numeric")
- **z**: matrix of covariates, not including the modifier (class "matrix")
- **df_m**: degrees of freedom for modifier basis (class "numeric")
- **df_l**: degrees of freedom for exposure time basis (class "numeric")
- **penalize**: TRUE to penalize model (class "logical")
- **pen_fn**: if penalizing, can specify "ps" for penalized B-splines or "cr" for cubic regression splines with penalties on second derivatives
- **mod_args**: a list of additional arguments for the spline function (must be named by argument)
- **lag_args**: a list of additional arguments for the spline function (must be named by argument)
- **fit_fn**: specify "gam" to use the *gam* function for data sets that are not very large, and specify "bam" to use the *bam* function for data sets that are very large. Default will fit using *gam*. (class "character")
- **model_type**: "linear" for a DLIM with linear interaction, "quadratic" for a DLIM with quadratic interaction, "standard" for a DLIM with splines (class "character")
- **ID**: group identifier for random intercept, only supported for penalized models
- **...**: Other arguments to pass to model fitting function

Value

This function returns a list that is an object of class "dlim" with the following components

- **cb**: cross-basis (class "matrix")
- **fit**: model object (class "lm", "glm", "gam")
- **modifiers**: modifying values (class "numeric")
- **call**: model call

See Also

Type `vignette('dlimOverview')` for a detailed description.

- `predict.dlim`
- `plot_cumulative`
- `plot_DLF`
Examples

```r
library(dlim)
data("ex_data")
dlim_fit <- dlim(y = ex_data$y,
               x = ex_data$exposure,
               modifier = ex_data$modifier,
               z = ex_data$z,
               df_m = 10,
               df_l = 10,
               method = "REML")
dlim_pred <- predict(dlim_fit,
                     newdata = 0.5,
                     type="CE")
```

---

**exposure**

*Exposure data set for simulation*

**Description**

Data set of PM 2.5 exposure history for 1000 individuals over 37 weeks

**Usage**

`exposure`

**Format**

A data frame of 1000 rows and 37 columns

**Source**

Data source??

**Examples**

```r
data(exposure) # lazy load
```
**ex_data**

*Example data set*

Description

Data set for examples

Usage

```r
ex_data
```

Format

List of response, exposure, modifiers, covariates

Source

Simulated

Examples

```r
data(ex_data) # lazy load
```

---

**model_comparison**

*Model Comparison*

Description

Compare models to test for interaction

Usage

```r
model_comparison(fit, null = "DLM", x, B, conf.level = 0.95)
```

Arguments

- `fit`: dlim object (must be fit with REML)
- `null`: "DLM", "linear" to specify the null model
- `x`: exposure
- `B`: number of bootstrap samples
- `conf.level`: The confidence level (class "numeric")

Value

The function returns a decision to either reject or fail to reject the null model
plot_cumulative

Description
Plot estimated distributed lag function values from a DLIM object, can also compare those of a DLM.

Usage
plot_cumulative(
  new_modifiers,
  mod_fit,
  dlm_fit = NULL,
  mod_name = NULL,
  mod_trans = NULL,
  link_trans = NULL
)

Arguments
- new_modifiers: a vector of new modifier values for prediction (class "numeric")
- mod_fit: DLIM model object (class "dlim")
- dlm_fit: a list containing a crossbasis object from the dlnm package as the first element and a DLM model object as the second element (class "list")
- mod_name: modifier name (character)
- mod_trans: if modifiers are transformed, specify back transformation function (class "character")
- link_trans: if family for glm is not Gaussian, specify back transformation to undo link function (class "character")

Value
This function returns a ggplot for cumulative effects, including for a DLM if specified

See Also
dlim
Type vignette('dlimOverview') for a detailed description.
**plot_DLF**  

*Plot Cumulative Effects*

**Description**

Plot estimated cumulative effects from a DLIM object, can also compare estimated cumulative effects between a DLM and DLIM

**Usage**

```r
cpylot_DLF(
    new_modifiers,
    mod_fit,
    mod_name,
    dlm_fit = NULL,
    plot_by,
    time_pts = NULL,
    mod_trans = NULL,
    link_trans = NULL
)
```

**Arguments**

- `new_modifiers`: a vector of new modifier values for prediction (class "numeric")
- `mod_fit`: DLIM model object (class "dlim")
- `mod_name`: modifier name that follows variable name nomenclature (class "character")
- `dlm_fit`: a list containing a crossbasis object from the `dlm` package as the first element and a DLM model object as the second element (class "list")
- `plot_by`: choose to create plots for particular modifier values, "modifier", or particular time points, "time", (class "character")
- `time_pts`: a set of time points if plotting by time (class "numeric")
- `mod_trans`: if modifiers are transformed, specify back transformation function (class "character")
- `link_trans`: if family for `glm` is not Gaussian, specify back transformation to undo link function (class "character")

**Value**

This function returns a ggplot for point-wise effects isolated by either time points or modifier, including a DLM if specified

**See Also**

- `dlim`

Type `vignette('dlimOverview')` for a detailed description.
**predict.dlim**  

**DLIM Predictions**

**Description**

Predicted values based on a `dlim` object.

**Usage**

```
## S3 method for class 'dlim'
predict(
  object,
  newdata = NULL,
  type = c("DLF", "CE", "response"),
  conf.level = 0.95,
  ...
)
```

**Arguments**

- **object**  
an object of class "dlim"
- **newdata**  
a vector of new modifier values for prediction (class "numeric")
- **type**  
Type of prediction. "DLF" for the estimated distributed lag functions, "CE" for cumulative effects, "response" for fitted values, or any combination of these in a vector (class "character")
- **conf.level**  
The confidence level (class "numeric")
- **...**  
additional arguments affecting the predictions produced

**Value**

This function returns a list of 3 elements:

- **est_dlim**  
cumulative and/or point-wise estimates, standard errors, and confidence intervals (class "list")
- **cb**  
cross-basis object (class "cross-basis")
- **model**  
model object (class "gam")

**See Also**

- **dlim**

Type `vignette('dlimOverview')` for a detailed description.
Simulated DLIM Predictions

Description

This function estimates cumulative and non-cumulative lag/modifier coefficients from a model in which the response is regressed on a cross-basis generated by the `cross_basis()` function.

Usage

```r
## S3 method for class 'sim_dlim'
predict(object, newdata = NULL, type = c("DLF", "CE", "response"), ...)
```

Arguments

- `object`: an object of class "dlim"
- `newdata`: vector of modifiers for inference (class "numeric")
- `type`: Type of prediction. "response" for predicted responses, "DLF" for the estimated distributed lag functions, "CE" for cumulative effects (class "character")
- `...`: additional arguments affecting the predictions produced

Value

This function returns a list of 4 or 7 elements:

- `est_dlim`: `est_dlim` element from `predict.dlim` (class "list")
- `cb`: `cross_bais` from object (class "cross-basis")
- `fit`: `fit` from object (class "lm", "glm", "gam")
- `true_betas`: `true_betas` from object (class "matrix")
- `cb_dlm`: `cb_dlm` from object (class "crosspred")
- `model_dlm`: `model_dlm` from object (class "lm", "glm", "gam")
- `est_dlm`: cumulative and/or point-wise estimates, standard errors, and confidence intervals for the DLM (class "list")

See Also

- `predict.dlim`

Type `vignette('dlimOverview')` for a detailed description.
**print.dlim**  
*Print DLIM Information*

**Description**

prints information about an object of class `dlim`

**Usage**

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

**Arguments**

- `x` a `dlim` object
- `...` further arguments passed to or from other methods

**Value**

This function returns information about an object of class `dlim`

**See Also**

Type `vignette('dlimOverview')` for a detailed description.

---

**sim_data**  
*Simulate Data*

**Description**

Simulate data to use with the `dlim` package. There are different effect modification scenarios to choose for simulation.

**Usage**

```r
sim_data(
  x,
  L = NULL,
  modifiers,
  noise = 1,
  type = 2,
  SNR,
  ncovariates = 0,
  gamma = 1
)
```
**Arguments**

- **x**: a time series vector of length \( n \) or matrix of lagged exposures for \( n \) individuals (class "numeric", "matrix")
- **L**: a vector of length 1 containing the number of lag terms. This is required if \( x \) is a vector, and is not used if \( x \) is a matrix (class "numeric")
- **modifiers**: vector of length \( n \) containing modifying values (class "numeric")
- **noise**: a vector of length 1 containing the standard deviation for a normal distribution with mean 0 used to add noise to the simulated response values. Must provide if SNR is not provided (class "numeric")
- **type**: a vector containing the number 1, 2, 3, or 4 for simulation modification type: none, linear, non-linear shift, non-linear shift with linear scale (class "numeric")
- **SNR**: The signal-to-noise ratio. If SNR is provided, but noise is not, noise is reset to be the standard deviation of the response, before adding noise. (class "numeric")
- **ncovariates**: number of covariates to add to the model, numeric vector of length 1.
- **gamma**: True coefficient for the main effect of the modifier (class "numeric")

**Value**

This returns a list of 8 items:

- **x**: a lagged exposure matrix. If \( x \) was a matrix, it is unchanged. (class "matrix")
- **L**: a numeric vector of length 1 containing the number of lag terms (class "numeric")
- **modifiers**: the modifiers argument (class "numeric")
- **y**: a numeric vector of length \( nrow(x) \) containing the perturbed simulated response values. (class "numeric")
- **betas**: a matrix containing true coefficients for each lag/modifier combination, with each row representing a lag and each column a modifier (class "matrix")
- **betas_cumul**: a numeric vector of length \( L+1 \) containing cumulative true coefficients for the lag terms, summed over modifiers (class "numeric")
- **Z**: covariates (class "matrix")
- **gammas**: true coefficients for the covariates (class "numeric")

**See Also**

- **sim_dlim**

Type `vignette('dlimOverview')` for a detailed description.
sim_dlf
Simulate Distributed Lag Functions

Description

generate true distributed lag function values for a given type of simulation

Usage

sim_dlf(L, modifiers, type)

Arguments

L Number of lags minus 1
modifiers Vector of modifiers
type Effect modification simulation type: 1 is no modification, 2 is linear scale modification, 3 is non-linear shift modification, 4 is types 2 and 3 combined

Value

This function returns the true distributed lag function values (class "numeric")

See Also

sim_data
type vignette('dlimOverview') for a detailed description.

sim_dlim
Fit DLIM for simulation

Description

Fit DLIM for simulation

Usage

sim_dlim(
  data,
  df_m,
  df_l,
  penalize = TRUE,
  pen_fn = "ps",
  mod_args = NULL,
  lag_args = NULL,
  fit_dlm = FALSE,
model_type = "standard",
...
)

Arguments

data output from sim_data
df_m degrees of freedom for modifiers
df_l degrees of freedom for lags
penalize True to penalize model
pen_fn if penalizing, can specify "ps" for penalized B-splines or "cr" for cubic regression splines with penalties on second derivatives
mod Args a list of additional arguments for the spline function (must be named by argument)
lag_args a list of additional arguments for the spline function (must be named by argument)
fit_dlm True to additionally fit dlm for comparison
model_type "linear" for a DLIM with linear interaction, "quadratic" for a DLIM with quadratic interaction, "standard" for a DLIM with splines
... arguments to pass to model fitting function

Value

This function returns an object of class "sim_dlim"

cb DLIM cross-basis (class "cross-basis")
fit DLIM model fit (class "lm", "glm", "gam")
cb_dlm DLM cross-basis (class "crossbasis")
model_dlm DLM model fit (class "lm", "glm", "gam")
true_betas true linear effect of the exposure on the response for each individual and time point (class "matrix")
modifiers modifiers from numeric
data data (class "list")

See Also

dlim
sim_data

Type vignette('dlimOverview') for a detailed description.
Summary of object of class dlim

Usage

## S3 method for class 'dlim'
summary(object, ...)

Arguments

- object: a dlim object
- ...: additional arguments affecting the summary produced

Value

This function returns a summary for an object of class dlim

See Also

Type vignette('dlimOverview') for a detailed description.
Index

* datasets
  ex_data, 7
  exposure, 6
* package
  dlim-package, 2

  cross_basis, 2, 3
  dlim, 2, 4, 4, 8-10, 15
dlim-package, 2

  ex_data, 7
  exposure, 6

  model_comparison, 7

  plot_cumulative, 2, 5, 8
  plot_DLF, 2, 5, 9
  predict.dlim, 2, 5, 10, 11
  predict.sim_dlim, 11
  print.dlim, 12

  sim_data, 12, 14, 15
  sim_dlf, 14
  sim_dlim, 13, 14
  summary.dlim, 16