Package 'dlm'

December 20, 2017

Version 0.1

Date 2017-12-1

title Distributed Lag Models for built environment applications

Description This package fits distributed lag models (DLMs) to describe how the association between the presence of built environment features and an outcome varies as a function of distance between locations for study participants and locations for environment features.

Author Andrew Whiteman <awhitem@umich.edu>, Jonggyu Baek <jongguri@umich.edu>, Brisa Sanchez <brisa@umich.edu>

Maintainer Andrew Whiteman <awhitem@umich.edu>

License GPL (>= 2)

Depends R (>= 3.0.2), lme4 (>= 1.1), ggplot2, methods

LinkingTo

Imports Matrix

Suggests devtools

RoxygenNote 6.0.1

RemoteType local

RemoteUrl /Users/AprilVKwong/Dropbox/software/dlm

RemoteSha NA

RemoteBranch master

RemoteUsername asw221

RemoteRepo dlm

R topics documented:

`dlm-package` *dlm: Distributed lag models in R using lme4*

Description

This package fits the distributed lag models (DLMs) described by Baek et al (2016) and Baek et al (2017), which estimate the association between the presence of built environment features and an outcome as a function of distance between the locations for study participants and locations for environment features or community resources. These models circumvent the need to pre-specify a radius within which to measure the availability of community resources. Distributed lag models have a long history in a variety of fields. For built environment research, we define the lagged exposure as the value of an environment feature between two radii, r_{l-1} and r_l from study locations, $l = 1, 2, \ldots, L$, where $r_0 = 0$; e.g., the lagged exposure is the number of convenience stores within "ring"-shaped areas around study participants residential address. The package supports generalized linear regression models, as well as generalized linear mixed models. In both instances, multiple lagged exposure covariates maybe included, as well as interactions between the lagged covariates and other categorical covariates (e.g., quartiles of age).

Let Y_{ij} be an outcome measured at location i at visit j, and $X_{ij}(r_{l-1}; r_l)$ be an environment feature measured during visit j within a ring-shaped area around location i between radii r_{l-1} and r_l; and r_L be the maximum distance around locations beyond which there is no association between the environment feature and the outcome. A typical unadjusted generalized linear mixed model that can be fitted in this version of the package is,

$$g(E(Y_{ij}|b_i)) = \beta_0 + \sum_{l=1}^{L} \beta(r_{l-1}; r_l) X(r_{l-1}; r_l) + W_{ij} b_i$$

where $g(\cdot)$ is a link function appropriate for the distribution of the outcome; β_0 represents an intercept; the association of the environment feature measured between radii r_{l-1} and r_l and the outcome is $\beta(r_{l-1}; r_l)$; and W_{ij} are covariates related to random effects, b_i (e.g., random intercepts and slopes). The coefficients $\beta(r_{l-1}; r_l)$ are constrained to follow a smooth function of distance from the locations of interest; the constraint is imposed by modeling the coefficients using smoothing splines. Other models could be used, although smoothing splines are the only supported option at this time.

The model easily simplifies to generalized linear regression modes (e.g., when there is only one visit), and can be extended in the following directions. Adjustment covariates can be easily included. In addition, interaction terms between covariates and the DL covariates are also supported. For example, terms such as: $\sum_{l=1}^{L} \theta(r_{l-1}; r_l) X(r_{l-1}; r_l) Z_i$, where Z_i is another covariate, can be included. The interaction coefficients $\theta(r_{l-1}; r_l)$ have the usual interpretation, but the magnitude of the interaction can vary over distance from locations of interest; \theta(r_l-1; r_l) are also constrained using smoothing splines. Finally, weighted regression models are also supported.

We assume the user has calculated distances from every participant's location to every community resource/feature. The distances can be network distances or Euclidian distances. Those distances are then used to calculate the distributed lag covariates, X(r_l-1; r_l), by specifying L and the radii r_l, $l = 1, 2, \ldots, L$. See Baek et al (2016) for guidance on choosing L and r.

The package include a series of functions to pass formulas and data to **lme4**, which is used for estimation of the DLM. All those functions are documented in this manual, although a typical user will primarily interact with XXX, xxx, and xxx. For example:

References

Baek J, Sanchez BN, Berrocal VJ, & Sanchez-Vaznaugh EV (2016) Epidemiology 27(1):116-24. (PubMed)

Baek J, Hirsch JA, Moore K, Tabb LP, et al. (2017) Epidemiology 28(3):403-11. (PubMed)

Bates D, Maechler M, Bolker BM, & Walker SC (2015) Fitting linear mixed-effects models using lme4. J Stat Softw 67(1). (jstatsoft.org)

`basis` *Basis vector sets*

Description

Construct a set of basis vectors based on the distances between input points. Typical usage relies on calling basis application functions, like `cr` (e.g. in `dlm` model formulas); users should not often have to interact with `basis` directly

Usage

```
basis(x, center = TRUE, scale = FALSE, .fun = NULL, ...)
```

Arguments

`x`	a set of points to measure distances between; the resultant distance matrix will be decomposed into a set of basis vectors
`center`	if `TRUE` (the default), parameter `x` will be mean centered prior to computing distances. Otherwise, if given a `numeric` value, `x` will be centered at `center`
`scale`	if `TRUE` (default = `FALSE`), parameter `x` will be scaled by `sd(x)`. Otherwise, if given a `numeric` value, `x` will be scaled by `scale`
`.fun`	a function to compute distances between the values in `x`. The default is to compute pairwise cubed absolute distances. See Details
`...`	other parameters passed to `.fun`

Details

Alternative distance functions, `.fun`, may be specified, and error checking on the user's choice of `.fun` is deliberately missing. Proper candidates for `.fun` should return a (`length(x)` by `length(x)`) matrix, however, and values are typically non-negative.

In addition, new distance function definitions should follow the idiom:

```
function(x, y, ...)
  if (missing(y))  y <- x
  ...
```

The default value of `.fun` computes cubic radial distance, which amounts to `abs(outer(x, y, "-"))^3`; distance matrix decomposition follows Rupert, Wand, and Carroll (2003). In particular, once `x` and `.fun` are chosen, define distance matrix `C_1 = .fun(x, ...)`, and let

$$C_0 = [1^{(n\times 1)}, x]$$

$$C_1 = QR$$

$$M_1 = Q_{-(1:2)}$$

$$K_1 = C_1 M_1 (M_1^T C_1 M_1)^{-\frac{1}{2}}$$

where A_{-j} denotes a matrix A with column(s) j removed. Then distributed lag effects of interest are scaled by the matrix $\Omega = [C_0, K_1]$

Value

An object of class `LagBasis`

References

Rupert D, Wand MP, & Carroll RJ (2003) Semiparametric Regression. New York: Cambridge University Press.

See Also

`cr`, `dlm`

`changePoint` *Lag Coefficient Change Points*

Usage

```
changePoint(object, ...)
```

Value

An integer vector

cholfVar *Extract Cholesky factor of inverse Information matrix*

Usage

```
cholfVar(object, ...)
```

dlm *Distributed lag models*

Description

Fit distributed lag models using **lme4** to penalize smooth terms. Other random effects terms and generalized mixed models supported.

Usage

```
dlm(formula, data, subset, na.action, weights, offset,
    method = c("REML", "MLE"), family = gaussian(),
    control = list(), ...)
```

Arguments

formula	an object of class stats::formula: a symbolic description of the model to be fitted. See Details
data	an optional data frame, list, or environment containing the variables of the model to be fitted
subset	optional vector specifying a subset of observations to be used in the fitting process
na.action	optional function that indicates what should happen when the data contains NA's. The default is set by the na.action setting of base::options, usually stats::na.omit
weights	optional vector of weights to be used in the fitting process. Should be NULL or a numeric vector
offset	a known offset term to include in the model, e.g. for poisson() family models
method	algorithm used to fit the DLM. Partial matching and capitalization allowed. The default is "REML" for linear/gaussian(link = "identity") family models, and "MLE" otherwise
family	a description of the error distribution and link function to be used in the model. The default is gaussian(link = "identity"). See stats::family for possible family functions and details
control	either a list object with arguments to be passed to the lme4::lmerControl sequence, or the output of [g]lmerConrol directly
...	Additional parameters passed to lme4::lFormula

Details

Models are specified using typical **lme4** `formula` syntax with at least one set of lag terms returned by a given smoothing function (e.g. see `cr`). The smoothing function can be any that returns a `SmoothLag` basis object. See Examples for a basic call to `dlm` using the formula interface, and a cubic radial lag basis specified via `cr`.

Here, we consider models of the general form,

$$y_i = \alpha + \sum_{t=1}^{n^*} \beta(t)x(t) + \epsilon_i$$

where $t = 1, \ldots, n^*$ indexes a single set of lag coefficients, β. In general, the model can be extended to include other fixed effects, other sets of lag terms, and random effects. Above, $\beta(t)$ and $x(t)$ can be thought of as the average linear effect and concentration of events, respectively, occurring between two radii, r_{t-1}, and r_t (see References).

Value

An S4 object that inherits from `dlMod` and `lme4::merMod` containing the results of the fitted model. Many standard model summary methods are available for these object types

References

Baek J, Sanchez BN, Berrocal VJ, & Sanchez-Vaznaugh EV (2016) Epidemiology 27(1):116-24. (PubMed)

Baek J, Hirsch JA, Moore K, Tabb LP, et al. (2017) Epidemiology 28(3):403-11. (PubMed)

Bates D, Maechler M, Bolker BM, & Walker SC (2015) Fitting linear mixed-effects models using lme4. J Stat Softw 67(1). (jstatsoft.org)

See Also

`lme4::lmer`, `cr`, `dlMod`

Examples

```
data (simdata)

## Setup distance count matrix and corresponding lag distances
X <- as.matrix(simdata[, -(1:3)])
lag <- seq(0.1, 10, length.out = ncol(X))

fit <- dlm(Y ~ Age + Gender + cr(lag, X), data = simdata)
summary (fit)
```

dlMod *Distributed lag models*

Description

A fitted distributed lag model object. Inherits from **lme4**'s `merMod` so that most methods defined for this parent class should work seamlessly within **dlm** analysis

Slots

`resp` An `lme4::lmResp` object to store a (mixed) model response variable

`bases` A list of `LagBasis` objects corresponding to the unique set of lag bases used to fit the model

`index` A (named) integer vector providing the index of the basis set in `bases` corresponding to each lag term in the model

interpret.dlm *Interpret a DLM Formula*

Description

Given an appropriate DL model formula and data-frame, prepares the data to be fit by interpreting the formula and extracting available smooth lag terms.

Usage

```
interpret.dlm(formula, data, .names.func = function(n) paste("pseudoGroups",
  n, sep = ""))
```

Arguments

`formula`	a symbolic description of the model to be fitted. See `dlm` for details.
`data`	a model-frame containing the data for each term in the model. Should already be appropriately subset, etc.
`.names.func`	a function for creating names of dummy variables that act as placeholders for penalized spline terms in **lme4**'s setup. There should not be a need to alter this in normal use cases

Details

Users should not typically have to interact with `interpret.dlm` directly, but it may be useful for extensions.

Uses R's `stats::model.matrix` mechanisms to build and parse the random effects (or penalized) components of spline-lag terms in the model. The object returned is later passed to other **dlm** functions in order to fit the specified model.

Value

an S3 object of class "parsed.dlm" with list elements:

formula the formula passed to dlm

lme4.formula a reconstructed formula that is then passed to the **lme4** modular functions

model a data.frame returned by call to stats::model.frame

Bt a matrix of the random or penalized lag basis vectors, where each vector is a row. Stored as an object that inherits from Matrix::dMatrix

bases a list of all the unique bases represented in the formula. This may be <= the number of separate spline-lag terms. All elements should inherit from SmoothLag

lag.group an integer vector returned by parse.names where each unique integer corresponds to a separate spline-lag term. For lag term i, lag.group == i indexes the rows of Bt that correspond to the set of random or penalized basis vectors for that term

bi for "basis index." Each set of lag terms indexed in lag.group has a matching basis decomposition in bases. bi keeps track of that matching

---

LagBasis *Create and Manipulate Lag Basis Functions*

Description

S4 class object to store and query components of lag basis functions. User interface for creating this class can be found in basis.

Details

See basis for details of the decomposition

Slots

x original lag data. The set of points to generate the distance-decomposition basis from

x.center store the value the lag data was centered to

x.scale store the value the lag data was scaled by

C0 C_0 part of basis matrix

K1 K_1 part of basis matrix

dist.fun whatever distance function was used to compute the pointwise distances between the elements of x

References

Rupert D, Wand MP, & Carroll RJ (2003) Semiparametric Regression. New York: Cambridge University Press.

lagIndex *Extract list of indices of lag terms*

Usage

```
lagIndex(object, ...)
```

lme4.dlm *lme4.dlm*

Description

Fits an interpreted distributed lag model using **lme4** modular functions

Usage

```
lme4.dlm(parsed, family = gaussian(), control = list(), REML = FALSE, ...)
```

Arguments

parsed	an interpreted dlm formula object returned by interpret.dlm
family	a description of the error distribution and link function to be used in the model. The default is gaussian(link = "identity"). See stats::family for possible family functions and details
control	either a list object with arguments to be passed to the lme4::lmerControl sequence, or the output of [g]lmerConrol directly
REML	if TRUE and a linear dlm model is specified, lme4.dlm will use REML to fit the model. MLE will be use otherwise
...	other parameters to be passed to the **lme4** modular family functions

Details

Together with interpret.dlm, this function does the main grunt work for dlm. Given an interpreted model, lme4.dlm organizes the parsed data into the **lme4** modular functions to fit the model and returns the fit as an **lme4** object.

Value

an object that inherits from lme4::merMod containing a fitted model

makeDlMod *makeDlMod*

Description

Convert an appropriately fit model object into an object of class dlMod

Usage

```
## S3 method for class 'merMod'
makeDlMod(object, parsed, call, ...)
```

Arguments

object	a fitted model object
parsed	an interpreted dlm formula object returned by interpret.dlm
call	an optional matched function call

Value

an S4 object of class dlMod

omega *Extract Lag Basis Matrix*

Description

Extract lag basis matrix, Ω

Usage

```
omega(object, ...)

## S4 method for signature 'LagBasis'
omega(object, ...)
```

Details

omega is S4 generic.

Value

A square numeric matrix

Methods (by class)

- LagBasis: Method for "LagBasis" objects

parse.names *parse.names*

Usage

```
parse.names(base, names, .warn = TRUE)
```

plotDlm *Plot smoothed lag terms*

Description

Plot estimated lag-spline effects

Usage

```
plot(x, geom = c("pointrange", "line"), level = 0.95, scaled = TRUE, ...)
```

Value

a ggplot2 graphic object

predict.LagBasis *Predict New Values for Fitted Lag Basis*

Description

Will update w/ description. Not yet implemented

Usage

```
## S3 method for class 'LagBasis'
predict(object, x, ...)
```

Arguments

object	A LagBasis object

scaleMat *Extract Distributed Lag Scale Matrix*

Description

Return lag coefficient scale matrix, S, such that the distributed lag coefficients fit by the model are obtained via the transformation $\beta = S\theta$. S should be invertable.

Usage

```
scaleMat(object, ...)

## S4 method for signature 'dlMod'
scaleMat(object, ...)
```

Arguments

object	a fitted model object

Details

scaleMat is S4 generic.

Value

A square numeric matrix

Methods (by class)

- dlMod: S4 Method for "dlMod" Objects

Sigma *Extract model coefficients variance matrix*

Usage

```
Sigma(object, ...)
```

simdata *Simulated Built Environment Data*

Description

Simulated Built Environment Data

Usage

```
data(simdata)
```

Format

The first column (`Y`) is the outcome variable, covariates `Gender` and `Age` come next, and are followed by 100 location description variables. Each column of these 100 location count variables corresponds to a distance lag equal to the values in `seq(0.1, 10, 0.1)`.

References

Baek J, Sanchez BN, Berrocal VJ, & Sanchez-Vaznaugh EV (2016) Epidemiology 27(1):116-24. (PubMed)

Baek J, Hirsch JA, Moore K, Tabb LP, et al. (2017) Epidemiology 28(3):403-11. (PubMed)

Bates D, Maechler M, Bolker BM, & Walker SC (2015) Fitting linear mixed-effects models using lme4. J Stat Softw 67(1). (jstatsoft.org)

Examples

```
data(simdata)
simdata[1:10, 1:5]
```

smoothing *Basis smoothing*

Description

Construct a set of basis vectors for distances between distributed lag points, and apply as a linear transformation of a concentration matrix.

Usage

```
cr(x, Z, ...)

sm(x, Z, ..., .fun = NULL)
```

Arguments

`x`	a vector of values to construct the basis from. Missing values are not allowed
`Z`	a covariate matrix (or object that can be coerced to a `matrix`) to apply the linear basis transformation to. `length(x)` should be the same as `ncol(Z)`
`...`	arguments to be passed to `basis`

Details

These functions are little more than convenient wrappers to the function basis and the SmoothLag class constructor. They are intended to simplify the task of specifying lag terms in a model formula. The functions computes a set of basis vectors for parameter x and applies this basis as a linear transformation of the covariate/concentration matrix parameter, Z. For example, if Z is the identity matrix, the model fit will simply be the natural cubic spline of x.

Note that other basis extensions should always return an object that inherits from SmoothBasis

Value

An S4 object of class SmoothBasis.

Functions

- cr: natural cubic radial basis spline
- sm: user-defined smoothing

References

Rupert D, Wand MP, & Carroll RJ (2003) Semiparametric Regression. New York: Cambridge University Press.

See Also

basis, SmoothBasis

Examples

```
## load simulated data set and extract concentration matrix
data (simdata)
Conc <- simdata[, -(1:3)]  # First columns are Y, Age, and Gender

## radial lag (distance) each concentration was measured at
x <- seq(0.1, 10, length.out = ncol(Conc))
crb <- cr(x, Conc)
```

SmoothLag *Lag Matrix With Applied Smoothing*

Description

An S4 object for representing lag covariates and storing details about the basis smoothing. Intended for use within the dlm modeling framework to assist extraction of basis components treated as "fixed" and "random" effects. Inherits from matrix.

Slots

`basis` A `LagBasis` smoothing object containing details about the lag and the smoothing parameters used

`.Data` Contains the "fixed effects" components of the smoothed lag function. This scheme is intended to work conveniently with `stats::model.matrix`

`random` Contains the random effects or penalized components of the smoothed lag function

`terms` Character vector containing the function name and deparsed arguments from whatever smoothing function generated the `SmoothLag` object

---

vcoef *Vectorized coefficients*

Description

Extract fixed and random effects coefficient vector, $(\beta, b)'$ from a fitted `dlMod` object

Usage

```
vcoef(object, ...)

## S4 method for signature 'dlMod'
vcoef0(object, scaled = TRUE, ...)

## S4 method for signature 'dlMod'
vcoef(object, scaled = TRUE, ...)
```

Value

A numeric vector: with variable names in the case of `vcoef`

Methods (by class)

- `dlMod`: S4 Method for "`dlMod`" Objects
- `dlMod`: S4 Method for "`dlMod`" Objects

Index