Package 'lmeSplines'

July 22, 2025

Version 1.1-12

Title Add Smoothing Spline Modelling Capability to `nlme`

Author Rod Ball <rod.ball@scionresearch.com>

Maintainer Andrzej Galecki <agalecki@umich.edu>

Description Adds smoothing spline modelling capability to nlme. Fits smoothing spline terms in Gaussian linear and nonlinear mixed-effects models.

Depends nlme(>= 3.1-29)

License GPL (>= 2)

Repository CRAN

NeedsCompilation no

Date/Publication 2022-05-02 13:30:02 UTC

Contents

approx.Z 1
smspline 2
smSplineEx1 5

Index **6**

`approx.Z` *Interpolating in smoothing spline Z-matrix columns*

Description

Function to interpolate the Z-matrix for LME smoothing spline fits from one set of values of the `time` covariate to another.

Usage

```
approx.Z(Z, oldtimes, newtimes)
```

Arguments

Z	Z-matrix with rows corresponding to the sorted unique values of the time covariate
oldtimes	original (sorted) values for time covariate, corresponding to the rows of Z
newtimes	new (sorted) values for time covariate

Details

Uses linear interpolation of each column of the Z-matrix, regarded as a function of time, with the times given by oldtimes.

Value

A matrix corresponding to an interpolated spline matrix. Can be used with smspline() for fitting LME splines with random effects corresponding to a different set of values of the time covariate than those represented in the data, or as part of the 'newdata' argument prediction from an existing model, to obtain predictions at points not represented in the data using predict.lme().

Note

Linear interpolation works well here because the spline basis functions are approximately piecewise linear.

Author(s)

Rod Ball <rod.ball@scionresearch.com> https://www.scionresearch.com/

See Also

smspline nlme

Examples

```
times1 <- 1:10
Zt1 <- smspline(~ times1)
times2 <- seq(1,10,by=0.1)
Zt2 <- approx.Z(Zt1,oldtimes=times1,newtimes=times2)
```

---

smspline *Smoothing splines in NLME*

Description

Functions to generate matrices for a smoothing spline covariance structure, to enable fitting smoothing spline terms in LME/NLME.

Usage

```
smspline(formula, data)
smspline.v(time)
```

Arguments

formula	model formula with right hand side giving the spline covariate
data	optional data frame
time	spline 'time' covariate to smooth over

Details

A smoothing spline can be represented as a mixed model (Speed 1991, Verbyla 1999). The generated Z-matrix from `smspline()` can be incorporated in the users's dataframe, then used in model formulae for LME random effects terms at any level of grouping (see examples). The spline random terms are fitted in LME using an identity `'pdMat'` structure of the form `pdIdent(~Z - 1)`. The model formulation for a spline in time (`t`) is as follows (Verbyla 1999):

$$y = X_s\beta_s + Z_s u_s + e$$

where $X_s = [1|t]$, $Z_s = Q(t(Q)Q)^{-1}$, and u_s $N(0, G_s)$, is a set of random effects. We transform the set of random effects u_s to independence with $u_s = Lv_s$, where

$$v_s\ N(0, I\sigma_s^2)$$

is a set of independent random effects. The Z-matrix is transformed accordingly to $Z = Z_s L$, where L is the lower triangle of the Choleski decomposition of G_s.

The function `smspline.v()` is called by `smspline()`, and can be used to access the matrices X_s, Q, G_s. See Verbyla (1999) for further information.

Value

For `smspline()`, a Z-matrix with the same number of rows as the data frame. After fitting, the LME model output gives a standard deviation parameter for the random effects, estimating σ_s. The smoothing parameter from the penalised likelihood formulation is

$$\lambda = \sigma^2/\sigma_s^2$$

For `smspline.v()`, a list of the form

Xs	X-matrix for fixed effects part of the model
Zs	Z-matrix for random effects part of the model
Q, Gs, R	Matrices Q, G_s, R associated to the mixed-model form of the smoothing spline.

Note

The time points for the smoothing spline basis are, by default, the unique values of the time covariate. This is the easiest approach, and model predictions at the fitted data points, can be obtained using `predict.lme`. By interpolation, using `approx.Z`, the Z-matrix can be obtained for any set of time points and can be used for fitting and/or prediction. (See examples). Synopsis:data$Z <- smspline(formula1, data); fit <-lme(formula2, data, random= ...)

Author(s)

Rod Ball <rod.ball@scionresearch.com> https://www.scionresearch.com/

References

The correspondence between penalized likelihood formulations of smoothing splines and mixed models was pointed out by Speed (1991). The formulation used here for the mixed smoothing spline matrices are given in Verbyla (1999). LME/NLME modelling is introduced in Pinheiro and Bates (2000).

Pinheiro, J. and Bates, D. (2000) Mixed-Effects Models in S and S-PLUS Springer-Verlag, New York.

Speed, T. (1991) Discussion of "That BLUP is a good thing: the estimation of random effects" by G. Robinson. Statist. Sci., 6, 42–44.

Verbyla, A. (1999) Mixed Models for Practitioners, Biometrics SA, Adelaide.

See Also

approx.Z nlme

Examples

```
# smoothing spline curve fit
data(smSplineEx1)
# variable `all' for top level grouping
smSplineEx1$all <- rep(1,nrow(smSplineEx1))
# setup spline Z-matrix
smSplineEx1$Zt <- smspline(~ time, data=smSplineEx1)
fit1s <- lme(y ~ time, data=smSplineEx1,
    random=list(all=pdIdent(~Zt - 1)))
summary(fit1s)
plot(smSplineEx1$time,smSplineEx1$y,pch="o",type="n",
     main="Spline fits: lme(y ~ time, random=list(all=pdIdent(~Zt-1)))",
     xlab="time",ylab="y")
points(smSplineEx1$time,smSplineEx1$y,col=1)
lines(smSplineEx1$time, smSplineEx1$y.true,col=1)
lines(smSplineEx1$time, fitted(fit1s),col=2)

# fit model with cut down number of spline points
times20 <- seq(1,100,length=20)
Zt20 <- smspline(times20)
smSplineEx1$Zt20 <- approx.Z(Zt20,times20,smSplineEx1$time)
fit1s20 <- lme(y ~ time, data=smSplineEx1,
    random=list(all=pdIdent(~Zt20 - 1)))
# note: virtually identical df, loglik.
anova(fit1s,fit1s20)
summary(fit1s20)

# model predictions on a finer grid
times200 <- seq(1,100,by=0.5)
pred.df <- data.frame(all=rep(1,length(times200)),time=times200)
```

```
pred.df$Zt20 <- approx.Z(Zt20, times20,times200)
yp20.200 <- predict(fit1s20,newdata=pred.df)
lines(times200,yp20.200+0.02,col=4)


# mixed model spline terms at multiple levels of grouping
data(Spruce)
Spruce$Zday <- smspline(~ days, data=Spruce)
Spruce$all <- rep(1,nrow(Spruce))
# overall spline term, random plot and Tree effects
spruce.fit1 <- lme(logSize ~ days, data=Spruce,
                   random=list(all= pdIdent(~Zday -1),
                     plot=~1, Tree=~1))
# try overall spline term plus plot level linear + spline term
spruce.fit2 <- lme(logSize ~ days, data=Spruce,
                   random=list(all= pdIdent(~Zday - 1),
                     plot= pdBlocked(list(~ days,pdIdent(~Zday - 1))),
                     Tree = ~1))
anova(spruce.fit1,spruce.fit2)
summary(spruce.fit1)
```

`smSplineEx1` *Simulated data about a smooth curve*

Description

Simulated data to demonstrate smoothing spline curve fitting with `smspline` and `lme`

Usage

```
data(smSplineEx1)
```

Format

A data frame with 100 observations on the following 4 variables.

time time covariate

y simulated response values

y.true true response values

Details

100 data points were simulated about the curve $y = 10 - 6 * \exp(-4t/100)$, with iid normal random errors with standard deviation 1.

Examples

```
data(smSplineEx1)
```

Index

∗ **datasets**
    smSplineEx1, 5
∗ **manip**
    approx.Z, 1
∗ **models**
    smspline, 2
∗ **regression**
    smspline, 2
∗ **smooth**
    smspline, 2

approx.Z, 1, *3, 4

lme, *5*

nlme, *2*, *4*

predict.lme, *3*

smspline, *2*, 2, *5*
smSplineEx1, 5