

# Package ‘fcr’

March 13, 2018

**Title** Functional Concurrent Regression for Sparse Data

**Version** 1.0

**Author** Andrew Leroux [aut, cre],  
Luo Xiao [aut, cre],  
Ciprian Crainiceanu [aut],  
William Checkly [aut]

**Maintainer** Andrew Leroux <aleroux2@jhu.edu>

**Description** Dynamic prediction in functional concurrent regression with an application to child growth. Extends the `pffr()` function from the 'refund' package to handle the scenario where the functional response and concurrently measured functional predictor are irregularly measured. Leroux et al. (2017), *Statistics in Medicine*, <doi:10.1002/sim.7582>.

**Depends** R (>= 3.2.4), face (>= 0.1), mgcv (>= 1.7), fields (>= 9.0)

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.0.1

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2018-03-13 15:45:22 UTC

## R topics documented:

fcr-package . . . . .	2
content . . . . .	2
fcr . . . . .	3
plot.fcr . . . . .	6
predict.fcr . . . . .	7
<b>Index</b>	<b>8</b>

---

fcr-package	<i>Dynamic prediction in functional concurrent regression with sparse functional covariates</i>
-------------	---

---

### Description

This package contains the functions for fitting dynamic functional concurrent regression with sparse data.

### Notation

Let  $y_{ij}$  denote some outcome measured at  $t_{ij}$  on the functional domain (e.g. time) for subject  $i$  at observation  $j$ . We focus on fitting models of the form

$$y_{ij} = f_0(t_{ij}) + f_1(t_{ij})X_{ij} + \dots + b_i(t_{ij}) + \epsilon_{ij}$$

### Estimation

Estimation is performed using an iterative procedure described in Leroux et. al (2017). Initially, a model is fit without  $b_i(t_{ij})$ . Using the residuals from this initial fit, the covariance function is estimated. The model is then re-fit using this covariance function. This procedure can be iterated as many times as desired.

### References

Leroux A, Xiao L, Crainiceanu C, Checkley W (2017). Dynamic prediction in functional concurrent regression with an application to child growth.

---

content	<i>Example dataset</i>
---------	------------------------

---

### Description

Simulated data from the CONTENT dataset. Data contains information on child growth as measured by WHO defined Z-scores as well as gender.

### Format

A dataframe with 8 variables:

**Y** Observed HAZ score

**Ytrue** True HAZ score

**waz.true** True WAZ score

**waz** Observed WAZ score

**Male** Sex. 1 if male, 0 if female.

**argvals** time of observations standardized to be in the interval [0,1]

**subj** Subject ID

**include** Indicator for out of sample prediction used in the vignette

## References

Jaganath D, Saito M Gimán RH Queirox DM, Rocha GA, Cama V, Cabrera L, Kelleher D, Windle HJ, Crabtree JE, Jean E, Checkley W. First Detected *Helicobacter pylori* Infection in Infancy Modifies the Association Between Diarrheal Disease and Childhood Growth in Peru. *Helicobacter* (2014); 19:272-297.

---

 fcr

*Fit Functional Concurrent Regression*


---

## Description

This function implements functional concurrent regression for sparse functional responses with both functional and scalar covariates. This function is a wrapper for mgcv's [gam/bam](#).

## Usage

```
fcr(formula, argvals, subj, argvals.new = NULL, data = NULL, niter = 1,
    sp = FALSE, nPhi = NULL, use_bam = FALSE, discrete = FALSE,
    face.args = list(knots = 12, lower = -3, pve = 0.95), ...)
```

## Arguments

formula	formula will accept any input formula which is valid for <a href="#">gam</a> . The formula should only include terms not associated with the random function intercept $b_i(t_{ij})$ . See Examples.
argvals	a string indicating the functional domain variable name in data
subj	a string indicating the unique subject identifier name in data
argvals.new	new values of the functional domain to predict using <a href="#">face.sparse</a> , optional if one desires to predict at points of the functional domain not included in the data fitting procedure, they must be supplied in this argument.
data	dataframe including all variables of interest. Must not have any missing data for variables used in model fitting. data must also not contain any variables named: "g", "phi" followed by any numbers, or "sp" followed by any numbers. These names are reserved for the fitting procedure.
niter	number of times to iterate the covariance estimation
sp	logical argument indicating whether smoothing parameters for random effects should be supplied to <a href="#">gam</a> or <a href="#">bam</a> using estimates from <a href="#">face.sparse</a> (TRUE), or whether smoothing parameters for random effects should be estimated by mgcv (FALSE). Defaults to FALSE.

nPhi	number of random effects to include in final model (i.e. number of eigenfunctions of the covariance function). Default value (NULL) results in the use of all estimated random effects.
use_bam	logical argument indicating whether to use <code>gam</code> or <code>bam</code> . For moderate or large number of eigenfunctions it is recommended to use <code>bam</code> .
discrete	logical argument indicating whether whether to supple <code>discrete = TRUE</code> argument to <code>bam</code> . This argument may reduce computation time, but is currently listed as “experimental”. Not available when <code>use_bam = FALSE</code> . Defaults to <code>FALSE</code> .
face.args	list of arguments to pass to <code>face.sparse</code> . Can not pass the arguments “data”, “newdata”, “center” or “argvals.new” as these are determined by the procedure.
...	arguments to be passed to <code>mgcv::gam()/bam()</code>

## Details

The models fit are of the form

$$y = f_0(t_{ij}) + f_1(t_{ij})X_{ij} + \dots + b_i(t_{ij}) + \epsilon_{ij}$$

Note that this function will accept any valid formula for `gam/bam`. However, only the identity link function is available at this time. See the package vignettes for additional descriptions of dynamic prediction and the class of models fit by this function.

## Value

An object of class `fcr` containing five elements

**fit** An object corresponding to the fitted model from the `mgcv` package

**face.object** An object corresponding to the estimated covariance features

**runtime** Model fitting time

**argvals** Character scalar corresponding the name of the functional domain variable

**runtime** logical scalar corresponding to `sp` argument used in model fitting

## References

Jaganath D, Saito M, Gimán RH, Queirox DM, Rocha GA, Cama V, Cabrera L, Kelleher D, Windle HJ, Crabtree JE, Jean E, Checkley W. First Detected *Helicobacter pylori* Infection in Infancy Modifies the Association Between Diarrheal Disease and Childhood Growth in Peru. *Helicobacter* (2014); 19:272-297.

Leroux A, Xiao L, Crainiceanu C, Checkley W (2017). Dynamic prediction in functional concurrent regression with an application to child growth.

Xiao L, Li C, Checkley W, Crainiceanu C. Fast covariance estimation for sparse functional data. *Statistics and Computing*, (2017).

## Examples

```

data <- content
## smoothing parameters
k <- 12 # number of interior knots for fpca (results in k + 3 basis functions)
K <- 15 # dimension of smooth for time varying coefficients

## functional domain where we need predictions
tnew <- sort(unique(data$argvals))

#####
## Step 1: Smooth time-varying covariate ##
#####
dat.waz <- data.frame("y" = data$waz, "subj" = data$subj, argvals = data$argvals)
fit.waz <- face.sparse(dat.waz, newdata = dat.waz, knots = k, argvals.new = tnew)
data$wazPred <- fit.waz$y.pred

#####
## Step 2: Fit fcr ##
#####
fit <- fcr(formula = Y ~ s(argvals, k=K, bs="ps") +
           s(argvals, by=Male, k=K, bs="ps") +
           s(argvals, by=wazPred, bs="ps"),
           argvals = "argvals", subj="subj", data=data, use_bam=TRUE, argvals.new=tnew,
           face.args = list(knots=k, pve=0.99))

## plot covariance features
plot(fit, plot.covariance=TRUE)

## plot coefficient functions and qq plots for random effects
plot(fit)

#####
## Step 3: Prediction ##
#####
## data frames for in-sample and dynamic predictions
data_dyn <- data_in <- data

## change subject IDs to values not used in model fitting
## for dynamic prediction
data_dyn$subj <- data_dyn$subj + 1000

## make all observations beyond 0.5 NA in both data frames
## and dynamically predict the concurrent covariate in
## dynamic prediction
inx_na <- which(data_dyn$argvals > 0.5)

```

```

data_dyn$Y[inx_na] <- data_dyn$waz[inx_na] <- NA
data_dyn$wazPred <- predict(fit.waz,
                           newdata= data.frame("subj" = data_dyn$subj,
                                                "argvals" = data_dyn$argvals,
                                                "y" = data_dyn$Y))$y.pred

data_in$Y[inx_na] <- NA

## in sample and dynamic predictions on the same subjects
insample_preds <- predict(fit, newdata = data)
dynamic_preds  <- predict(fit, newdata = data_dyn)

```

---

plot.fcr

*Plotting an fcr model fit*


---

### Description

Plot method for `fcr`. Takes a fitted `fcr` object and plots either the features of the covariance function, or the smooth terms and qqplots for random effects. See `plot.gam` for further details.

### Usage

```

## S3 method for class 'fcr'
plot(x, plot.covariance = FALSE, ...)

```

### Arguments

<code>x</code>	object of class <code>fcr</code> .
<code>plot.covariance</code>	logical argument, indicates whether to plot features of the covariance function (correlation function, variance function, and eigenfunctions of the covariance function). If <code>FALSE</code> , will call <code>plot.gam</code> on the fitted <code>gam/bam</code> object. See <code>plot.gam</code> for additional details. Defaults to <code>FALSE</code> .
<code>...</code>	additional arguments to be passed to <code>plot.gam</code> .

### Value

If `plot.covariance` is `FALSE`, this function will silently return a list fo the data used to create the plots.

### Examples

```

## see examples in fcr

```

---

predict.fcr	<i>Prediction for fcr</i>
-------------	---------------------------

---

## Description

Predict method for [fcr](#)

## Usage

```
## S3 method for class 'fcr'
predict(object, newdata, type = "link", ...)
```

## Arguments

object	object of class <code>fcr</code> .
newdata	data frame including all predictors used in the model fitting procedure. missing values for the responses are OK. Missing covariate values will result in pairwise deletion with a warning message.
type	defaults to <code>link</code> (i.e. response). See <a href="#">predict.gam</a> for additional information.
...	additional arguments to be passed to <a href="#">predict.gam</a>

## Value

An object containing two elements

**dynamic\_predictions** Predictions corresponding to dynamic predictions (i.e. subject ids not included in the original fitting). Note that these predictions are slower and do not incorporate the estimated covariance between random effects and fixed effects in making predictions. This is different than the in sample predictions which account for this estimated covariance. See [predict.gam](#) for more information.

- fitted.values
  - y.pred fitted  $\hat{y}$
  - se.fit standard errors used to create confidence intervals for  $\hat{y}$ ,  $\sqrt{\text{var}(y)}$
  - se.fit.p standard errors used for creating prediction intervals for  $\hat{y}$ ,  $\sqrt{\text{var}(y) + \hat{\sigma}^2}$
  - random subject specific random effects  $b_i$
- scores matrix of BLUPs for subjects' eigenscores
- data newdata supplied to the function

**insample\_predictions** Predictions for subject ids included in the original fitting. This returns all output from the relevant [predict.gam/predict.bam](#) call.

## Examples

```
## see examples in fcr
```

# Index

`bam`, [3](#), [4](#)

`content`, [2](#)

`dynfcr` (`fcr-package`), [2](#)

`face.sparse`, [3](#), [4](#)

`fcr`, [3](#), [6](#), [7](#)

`fcr-package`, [2](#)

`gam`, [3](#), [4](#)

`plot.fcr`, [6](#)

`plot.gam`, [6](#)

`predict.bam`, [7](#)

`predict.fcr`, [7](#)

`predict.gam`, [7](#)