

# Package ‘fcomplete’

November 11, 2018

**Type** Package

**Date** 2017-08-14

**Title** Trajectory estimation for sparsely observed longitudinal data

**Version** 0.1.0

**License** GPL-3

**LazyData** TRUE

**RoxygenNote** 6.0.1

**Depends** R (>= 2.15.0),  
MASS,  
mvtnorm,  
fda,  
fpca,  
ggplot2,  
clusterGeneration,  
ElemStatLearn,  
tidyr

**Suggests** dplyr

**Roxygen** list(markdown = TRUE)

## R topics documented:

fregression . . . . .	2
fsimulate . . . . .	4
plot.fcomplete . . . . .	5

<b>Index</b>	<b>6</b>
--------------	----------

---

fregression	<i>Approximate low-rank processes from sparse longitudinal observations</i>
-------------	---

---

## Description

Method approximates a process from sparse observations. Suppose, that for a certain subject one or multiple observations are measured at some irregular timepoints. We assume that these are noise observations of some underlying process and we want to approximate this process for each individual.

## Usage

```
fregression(formula, data, covariates = NULL, bins = 51,
  method = c("fimpute", "fpcs", "mean"), lambda = c(0), maxIter = 1e+05,
  lambda.reg = 0, d = 7, K = NULL, K.reg = NULL, thresh = 1e-05,
  final = "soft", fold = 5, cv.ratio = 0.05, projection = "separate",
  verbose = 0, scale.covariates = TRUE)
```

## Arguments

formula	formula describing the linear relation between processes and indicating time and grouping variables. See details
data	data in the long format. Use <a href="#">fc.long2wide</a> and <a href="#">fc.wide2long</a> for conversions
bins	number of bins for matrix representation of the data
method	method for functional impute: <code>fpcs</code> for functional principal components, <code>mean</code> for mean impute and <code>fimpute</code> for functional impute
lambda	lambda for SVD regularization in functional impute
lambda.reg	lambda for SVD regularization in regression
d	dimensionality of the basis
K	upper bound of dimensionality for SVD regularization
K.reg	upper bound of dimensionality for regression
thresh	threshold for convergence in functional impute
final	should the final model use "hard" or "soft" impute after choosing the optimal lambda
fold	number of repetitions in cross-validation
projection	"joint" or "separate" (default). If multiple regressors are available project them jointly or separately
fold	how many folds in cross-validation

## Details

For a subject  $i$ , we observe  $Y^i(t), X_1^i(t), \dots, X_p^i(t)$  at irregular subject specific  $t \in t_1, \dots, t_p$ , where  $0 < t_j < T$ . We can bin the time interval  $[0, T]$  and represent each individual as a vector of fixed length with missing values. Let  $Y, X_1, \dots, X_p$  be such matrices. Columns correspond to timepoints and rows to subjects.

There are multiple methods of approximating the process  $Y$ , we can:

- regress  $Y$  on  $X_1, X_2, \dots, X_p$ , we can use sparse functional regression
- project each subject into latent space and impute  $Y, X_1, X_2, \dots, X_p$  simultaneously
- use only information from  $Y$ , we can use functional PCA method or functional impute.

Function `fregression` enables all three scenarios. Suppose `data` contains information in the long format, i.e. `data` is a matrix with  $p + 3$  columns, where `data[, 1]` is a subjectID, `data[, 2]` is time, `data[, 3]` is a value observation of  $Y$  and remaining columns are covariates  $X_1, \dots, X_p$ . Each row corresponds to one observation for one subject.

There are three possible formulas:

- $Y \sim \text{time} + X_1 + X_2 \mid \text{subjectID}$  executes functional regression
- $Y + X_1 + X_2 \sim \text{time} \mid \text{subjectID}$  executes dimensionality reduction
- $Y \sim \text{time} \mid \text{subjectID}$  executes functional impute or functional PCA depending on the choice of method parameter

## Value

Returns a list

- `fit` fitted matrix  $Y$
- meta results of cross-validation
- `u, d, v` svd of the underlying processes if the functional impute method has been chosen

In case of multidimensional SVD and simultaneous approximation of  $Y, X_1, X_2, \dots, X_p$ , `$fit` is a list of models for  $Y, X_1, X_2, \dots, X_p$ .

## References

James, Gareth M., Trevor J. Hastie, and Catherine A. Sugar. *Principal component models for sparse functional data*. *Biometrika* 87.3 (2000): 587-602.

## Examples

```
# SIMULATE DATA
simulation = fsimulate(seed = 1)
data = simulation$data
ftrue = simulation$ftrue
K = simulation$params$K

model.mean = fregression(Y ~ time | id, data,
                        method = "mean")
model.fpca = fregression(Y ~ time | id, data,
```

```

lambda = 0, K = c(3,4,5), thresh = 1e-7, method = "fpcs")

lambdas = c(2,3,4,5,6,8,10,12,15,20)
model.fimp = fregression(Y ~ time | id, data,
                        lambda = lambdas, thresh = 1e-5, final = "hard")
model.fcmp = fregression(Y + X1 + X2 ~ time | id, data, covariates,
                        lambda = lambdas, K = K, final = "hard")
model.freg = fregression(Y ~ U1 + U2 + time | id, data, model.fcmp$u,
                        lambda = lambdas, thresh = 1e-5,
                        lambda.reg = 0.1, method = "fpcs", K = K)

```

---

fsimulate

*Simulate data*


---

### Description

Function simulates data from the paper

### Usage

```

fsimulate(n = 100, d = 7, K = 3, dgrid = 51, clear = 0.85,
          noise.mag = 0.1)

```

### Arguments

n	number of observations
d	number of dimensions
K	true 'low dimension'
dgrid	size of the grid
clear	fraction of observations to remove
noise_mag	the magnitude of noise

### Details

The function simulates data as described in the paper. Generates observations  $X_1$ ,  $X_2$  and a  $Y = X_1 + X_2 + \text{noise}$

### See Also

[fregression](#)

---

plot.fcomplete	<i>Plots the fcomplete fit object</i>
----------------	---------------------------------------

---

**Description**

Plots the fcomplete fit object

**Usage**

```
## S3 method for class 'fcomplete'  
plot(model, rows = 1, true = NULL, test = NULL,  
      title = NULL, filename = NULL, ylim = NULL, width = 10, height = 10)
```

**Arguments**

n	number of observations
d	number of dimensions
K	true 'low dimension'
dgrid	size of the grid
clear	fraction of observations to remove
noise_mag	the magnitude of noise

**Details**

The function simulates data as described in the paper. Generates observations  $X_1, X_2$  and a  $Y = X_1 + X_2 + \text{noise}$

**See Also**

[fregression](#)

# Index

`fc.long2wide`, [2](#)  
`fc.wide2long`, [2](#)  
`fregression`, [2](#), [4](#), [5](#)  
`fsimulate`, [4](#)  
  
`plot.fcomplete`, [5](#)