Package 'FGLMtrunc'

December 22, 2025

Type Package

Title Truncated Functional Generalized Linear Models

Version 0.2.0

Date 2025-12-20

Author Xi Liu [aut],
Chau Tran [aut, cre],
Alexander Petersen [aut]

Maintainer Chau Tran <cbtr@ucdavis.edu>

Description An implementation of the methodologies described in Xi Liu, Afshin A. Divani, and Alexander Petersen (2022) <doi:10.1016/j.csda.2022.107421>, including truncated functional linear and truncated functional logistic regression models.

License GPL-2

Depends R (>= 3.6.0)

Imports foreach, glmnet, splines2

LinkingTo Rcpp, RcppArmadillo

Encoding UTF-8

RoxygenNote 7.1.2

Suggests knitr, xfun, rmarkdown

VignetteBuilder knitr

LazyData true

NeedsCompilation yes

Repository CRAN

Date/Publication 2025-12-22 18:10:14 UTC

Contents

`fglm_trunc` *Fit a truncated Functional Generalized Linear Model*

Description

Fit a truncated functional linear or logistic regression model using nested group lasso penalty. The solution path is computed efficiently using active set algorithm with warm start. Optimal tuning parameters (λ_s, λ_t) are chosen by Bayesian information criterion (BIC).

Usage

```
fglm_trunc(
  Y,
  X.curves,
  S = NULL,
  grid = NULL,
  family = c("gaussian", "binomial"),
  degree = 3,
  nbasis = NULL,
  knots = NULL,
  nlambda.s = 10,
  lambda.s.seq = NULL,
  precision = 1e-05,
  parallel = FALSE
)
```

Arguments

`Y`	n-by-1 vector of response. Each row is an observed scalar response, which is continous for family="gaussian" and binary (i.e. 0 and 1) for family="binomal".
`X.curves`	n-by-p matrix of functional predictors. Each row is an observation vector at p finite points on `[0,T]` for some `T>0`.
`S`	(optional) n-by-s matrix of scalar predictors. Binary variable should be coded as numeric rather than factor.
`grid`	A sequence of p points at which `X` is recorded, including both boundaries `0` and `T`. If not specified, an equally spaced sequence of length p between 0 and 1 will be used.
`family`	Choice of exponential family for the model. The function then uses corresponding canonical link function to fit model.
`degree`	Degree of the piecewise polynomial. Default 3 for cubic splines.
`nbasis`	Number of B-spline basis. If `knots` is unspecified, the function choose `nbasis - degree - 1` **internal** knots at suitable quantiles of `grid`. If `knots` is specified, the value of `nbasis` will be **ignored**.
`knots`	k **internal** breakpoints that define that spline.

nlambda.s	(optional) Length of sequence of smoothing regularization parameters. Default 10.
lambda.s.seq	(optional) Sequence of smoothing regularization parameters.
precision	(optional) Error tolerance of the optimization. Default 1e-5.
parallel	(optional) If TRUE, use parallel foreach to fit each value of lambda.s.seq. Must register parallel before hand, such as doMC or others.

Details

Details on spline estimator:

For an order q B-splines (q = degree + 1 since an intercept is used) with k internal knots 0 < t_1 <...< t_k < T, the number of B-spline basis equals q + k. Without truncation (λ_t=0), the function returns smoothing estimate that is equivalent to the method of Cardot and Sarda (2005), and optimal smoothing parameter is chosen by Generalized Cross Validation (GCV).

Details on family:

The model can work with Gaussian or Bernoulli responses. If family="gaussian", identity link is used. If family="binomial", logit link is used.

Details on scalar predictors:

FGLMtrunc allows using scalar predictors together with functional predictors. If scalar predictors are used, their estimated coefficients are included in alpha form fitted model.

Value

A list with components:

grid	The grid sequence used.
knots	The knots sequence used.
degree	The degree of the piecewise polynomial used.
eta.0	Estimate of B-spline coefficients η **without** truncation penalty.
beta.0	Estimate of functional parameter β **without** truncation penalty.
eta.truncated	Estimate of B-spline coefficients η **with** truncation penalty.
beta.truncated	Estimate of functional parameter β **with** truncation penalty.
lambda.s0	Optimal smoothing regularization parameter **without** truncation chosen by GCV.
lambda.s	Optimal smoothing regularization parameter **with** truncation chosen by BIC.
lambda.t	Optimal truncation regularization parameter chosen by BIC.
trunc.point	Truncation point δ where $\beta(t) = 0$ for $t \geq \delta$.
alpha	Intercept (and coefficients of scalar predictors if used) of truncated model.
scalar.pred	Logical variable indicating whether any scalar predictor was used.
call	Function call of fitted model.
family	Choice of exponential family used.

References

Xi Liu, Afshin A. Divani, and Alexander Petersen. "Truncated estimation in functional generalized linear regression models" (2022). *Computational Statistics & Data Analysis*.

Hervé Cardot and Pacal Sarda. "Estimation in generalized linear models for functional data via penalized likelihood" (2005). *Journal of Multivariate Analysis*.

See Also

bSpline from splines2 R package for usage of B-spline basis.

Examples

```
# Gaussian response
data(LinearExample)
Y_linear = LinearExample$Y
Xcurves_linear = LinearExample$X.curves
fit1 = fglm_trunc(Y_linear, Xcurves_linear, nbasis = 20, nlambda.s = 1)
print(fit1)
plot(fit1)
```

LinearExample *Simulated data for functional linear regression.*

Description

Randomly generated data with Gaussian responses for functional linear regression example follows Case I from Liu et. al. (2022).

Usage

```
data(LinearExample)
```

Format

List containing the following elements:

X.curves 200 by 101 matrix of functional predictors.

Y 200 by 1 numeric vector of Gaussian responses.

beta.true The true functional parameter β.

References

Xi Liu, Afshin A. Divani, and Alexander Petersen. "Truncated estimation in functional generalized linear regression models" (2022). *Computational Statistics & Data Analysis*.

LogisticExample *Simulated data for functional logistic regression.*

Description

Randomly generated data with Bernoulli responses for functional logistic regression example follows Case I from Liu et. al. (2022).

Usage

```
data(LogisticExample)
```

Format

List containing the following elements:

X.curves 200 by 101 matrix of functional predictors.

Y 200 by 1 numeric vector of Bernoulli responses.

beta.true The true functional parameter β.

References

Xi Liu, Afshin A. Divani, and Alexander Petersen. "Truncated estimation in functional generalized linear regression models" (2022). *Computational Statistics & Data Analysis*.

plot.FGLMtrunc *Plot functional parameters β from a* FGLMtrunc *object*

Description

Plot functional parameters β as a function of t for a fitted FGLMtrunc object.

Usage

```
## S3 method for class 'FGLMtrunc'
plot(x, include_smooth = TRUE, ...)
```

Arguments

x	fitted FGLMtrunc object
include_smooth	If TRUE, smoothing estimate without truncation of β is plotted.
...	additional plot arguments

Value

No return value.

`predict.FGLMtrunc` *Make predictions from* `FGLMtrunc` *fitted model*

Description

This function returns truncated estimate of linear predictors, fitted values, and functional parameter β for a fitted `FGLMtrunc` object.

Usage

```
## S3 method for class 'FGLMtrunc'
predict(
  object,
  newX.curves,
  newS = NULL,
  type = c("link", "response", "coefficients"),
  ...
)
```

Arguments

`object`	fitted `FGLMtrunc` object
`newX.curves`	Matrix of new values for functional predictors `X.curves`.
`newS`	Matrix of new values for scalar predictors `S`.
`type`	Type of prediction. For logistic regression (`family = "binomial"`), `type="link"` gives the linear predictors, which is log-odds, and `type="response"` gives the predicted probabilities. For linear regression (`family = "gaussian"`), both `type="link"` and `type="response"` give fitted values. For both linear regression and logistic regression, `type="coefficients"` gives truncated estimate of functional parameter β.
`...`	additional predict arguments (Not applicable for FGLMtrunc)

Value

Predictions depends on chosen `type`.

See Also

predict.glm.

`print.FGLMtrunc` *Print a* `FGLMtrunc` *object*

Description

Print a summary of truncation point of the fitted `FGLMtrunc` model.

Usage

```
## S3 method for class 'FGLMtrunc'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

`x`	fitted `FGLMtrunc` object
`digits`	significant digits in printout
`...`	additional print arguments

Details

Truncation point estimate of δ is printed.

Value

The fitted object is silently return.

Index