Package 'JOPSplus'

April 15, 2021

Type Package

Title Extensions to the JOPS package

Version 0.1.2

Author Paul Eilers and Brian Marx

Maintainer Paul Eilers <p.eilers@erasmusmc.nl>

Description The package JOPS is a companion to the book 'Practical Smmothing. The Joys of P-splines". This package contains later additions, not documented in the book

License GPL-2 | GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

R topics documented:

gpls *Partial least squares (multivariate calibration) for generalized linear models.*

Description

Partial least squares or PLS (iterative) fitting algorithm for GLMs.

Usage

```
gpls(
  y,
  X,
  components = "max",
  m_binomial = 0 * y + 1,
  r_gamma = 0 * y + 1,
  family = "gaussian",
  link = "default",
  max_threshold = 0.9,
  X_pred = NULL,
  y_pred = NULL,
  iter = 100
)
```

Arguments

y	a (glm) GLM response vector, usually continuous, binomial or count data.
X	a matrix of continuous regressor with nrow(X) == length(y) and ncol(X) = p , often a discrete digitization of a signal or histogram or time series.
components	integer number of latent variables, e.g. components = 4.
m_binomial	a vector of binomial trials having length(y); default is 1 vector for family = "binomial", NULL otherwise.
r_gamma	a vector of gamma shape parameters. Default is 1 vector for family = "Gamma", NULL otherwise.
family	the response distribution, e.g. "gaussian","binomial","poisson","Gamma" distribution; quotes are needed. Default is "gaussian".
link	the link function, one of "identity", "log", "sqrt", "logit", "probit", "cloglog", "reciprocal"; quotes are needed (default "identity").
max_threshold	if components is not set to an integer, then the number of components is set to capture this percent threshold of variability in X (default 0.90).
X_pred	a q-row matrix of external X (e.g. new signals) to yield external prediction.
y_pred	a vector of responses associated with X_pred which are used to calculate standard error of external prediction. Default is NULL.
iter	number of max iterations to construct latent variables.

Value

deviance	the deviance of gpls fit.
coef_X	a vector with length(p) of coefficients for centered and scaled X.
unscaled_coef_X	a vector with length(p) of unscaled coefficients for X.
mu	a vector with length(m) of estimated means.
eta_pred	a vector of length(q) of estimated linear predictors.
mu_pred	a vector of length(q) of estimated response (inverse link) predictors.
components	number of gpls components used in fit.
sep_pred	the cross-validation standard error of prediction for X_pred.
scale_vec	the scaling vector for X.
center_vec	the centering vector for X.
intercept_cs	the intercept term for centered and scaled X.
w_loadings	the matrix of loading vectors associated with component.
v_diag	the weight vectors of the glm.
glm_details	a list summary of glm fit using constructed gpls varialbles.
lv_coef	the coefficients associated with the construction of latent variables.
family	the family of the response.
link	the link function.

Author(s)

Brian Marx

References

Marx, B.D. (1996). Iteratively reweighted partial least squares estimation for generalized linear regression. Technometrics 38(4): 374-381.

Examples

```
library(JOPS)
library(JOPSplus)
# Get the data
library(fds)
data(nirc)
iindex <- nirc$x
X <- nirc$y
sel <- 50:650 # 1200 <= x & x<= 2400
X <- X[sel, ]
iindex <- iindex[sel]
dX <- diff(X)
diindex <- iindex[-1]
y <- as.vector(labc[1, 1:40]) # percent fat
oout <- 23
dX <- t(dX[, -oout])
y <- y[-oout]
fit1 <- gpls(y, dX, components=2, family = "gaussian", link = "identity", X_pred = dX)
plot(fit1, xlab = "Coefficient Index", ylab = "gpls coefficients")
```

```
title(main = "GPLS coefficient vector")
names(fit1)
predict(fit1, X_pred = dX[1:5,], type ='mu')
```

---

plot.gpls *Plotting function for* gpls

Description

Plotting function for generalized partial least squares coefficients (using gpls with class gpls).

Usage

```
## S3 method for class 'gpls'
plot(x, ..., xlab = "", ylab = "", col = "black", lty = 1)
```

Arguments

x	the gpls object, usually from gpls.
...	other parameters.
xlab	label for the x-axis, e.g. "my x" (quotes required).
ylab	label for the y-axis, e.g. "my y" (quotes required).
col	color.
lty	line type for plotting e.g. lty = 2.

Value

Plot	a plot of the gpls coefficent vector.

Author(s)

Brian Marx

#' @references Marx, B.D. (1996). Iteratively reweighted partial least squares estimation for generalized linear regression. Technometrics 38(4): 374-381.

Examples

```
library(JOPS)
library(JOPSplus)
# Get the data
library(fds)
data(nirc)
iindex <- nirc$x
X <- nirc$y
sel <- 50:650 # 1200 <= x & x<= 2400
X <- X[sel, ]
iindex <- iindex[sel]
dX <- diff(X)
diindex <- iindex[-1]
y <- as.vector(labc[1, 1:40]) # percent fat
oout <- 23
```

```
dX <- t(dX[, -oout])
y <- y[-oout]
fit1 <- gpls(y, dX, components=2, family = "gaussian", link = "identity", X_pred = dX)
plot(fit1, xlab = "Coefficient Index", ylab = "gpls coefficients")
title(main = "GPLS coefficient vector")
names(fit1)
```

`plot.ps2dvcsignal` *Plotting function for* `ps2DVCSignal`

Description

Plotting function for varying-coefficent signal regression P-spline smooth coefficients (using `ps2DVCSignal` with `class` `ps2dvcsignal`). Although se surface bands can be comuputed they are intentially left out as they are not interpretable, and there is generally little data to steer such a high-dimensional parameterization.

Usage

```
## S3 method for class 'ps2dvcsignal'
plot(x, ..., xlab = " ", ylab = " ", lty = c(1:8), col = c(1:8))
```

Arguments

`x`	the P-spline object, usually from `ps2DVCSignal`.
`...`	other parameters.
`xlab`	label for the x-axis, e.g. "my x" (quotes required).
`ylab`	label for the y-axis, e.g. "my y" (quotes required).
`lty`	linetype (vector) for plotting.
`col`	color (vector) for plotting.

Value

`Plot`	a plot of the estimated signal coefficient vector at a given (`t1_coef_loc`, `t2_coef_loc`).

Author(s)

Paul Eilers and Brian Marx

References

Eilers, P. H. C. and Marx, B. D. (2003). Multivariate calibration with temperature interaction using two-dimensional penalized signal regression. *Chemometrics and Intellegent Laboratory Systems*, 66, 159–174.

Eilers, P.H.C. and Marx, B.D. (2021). *Practical Smoothing, The Joys of P-splines.* Cambridge University Press.

Examples

```
library(JOPS)
library(JOPSplus)
fit1 <- ps2DVCSignal(y = rnorm(30), X = matrix(rnorm(30*100),30, 100), 1:100, t1_var = rnorm(30),
t2_var = rnorm(30), t1_coef_loc= c(0,1), t2_coef_loc=c(-1,0))
plot(fit1, xlab = "Coefficient Index", ylab = "Signal coeff", lty=c(1,2))
names(fit1)
```

`plot.psDensity` *Plotting function for* `psDensity`

Description

Plotting function for P-spline Density smooth with normal, Poisson, or binomial responses (`class psDensity`).

Usage

```
## S3 method for class 'psDensity'
plot(x, ..., xlab = "x", ylab = "density", col = "black", pch = 1)
```

Arguments

`x`	the P-spline object, usually from psDensity#' @param ... other parameters.
`xlab`	label for the x-axis.
`ylab`	label for the y-axis.
`col`	color for points.
`pch`	point character.

Value

`Plot`	a plot of the (inverse link) smoothed density.

Author(s)

Paul Eilers and Brian Marx

References

Eilers, P.H.C. and Marx, B.D. (2021). *Practical Smoothing, The Joys of P-splines.* Cambridge University Press.

Eilers, P.H.C., Marx, B.D., and Durban, M. (2015). Twenty years of P-splines, *SORT*, 39(2): 149-186.

Eilers, P.H.C. and Marx, B.D. (1996). Flexible smoothing with B-splines and

Examples

```
fitN = psDensity(rnorm(200), xl = -3.5, xr = 3.5, lambda = 10000)
plot(fitN, ylab= 'pdf', xlab = 'z-score')

#@examples
library(JOPS)
library(boot)
Year <- coal$date
xl <- min(Year)
xr <- max(Year)

# Poisson smoothing
nseg <- 50
bdeg <- 3
fit1 <- psDensity(Year, xl = 1845, xr = 1965, m=40, nseg, bdeg, pord = 3, lambda = 100)
plot(fit1, xlab = "Year", ylab = "Count")
```

`plot.psRandInt` *Plotting function for* `psRandInt`

Description

Plotting function for`psRandInt`, shared P-spline curves with random individual intercept offsets for gaussian, poisson, or binomial responses, updated penalties in mixed model style.

Usage

```
## S3 method for class 'psRandInt'
plot(x, ..., xlab = "", ylab = "", col = "black", lty = 1)
```

Arguments

`x`	the psRandInt object, usually from `psRandInt`.
`...`	other parameters.
`xlab`	label for the x-axis, e.g. "my x" (quotes required).
`ylab`	label for the y-axis, e.g. "my y" (quotes required).
`col`	color.
`lty`	line type for plotting e.g. `lty = 2`.

Value

`Plot`	a plot of the inverse link random intercept subject P-spline curves.

Author(s)

Brian Marx and Paul Eilers

References

Eilers, P.H.C. and Marx, B.D. (2021). *Practical Smoothing, The Joys of P-splines.* Cambridge University Press.

Eilers, P.H.C., Marx, B.D., and Durban, M. (2015). Twenty years of P-splines, *SORT*, 39(2): 149-186.

Eilers, P.H.C. and Marx, B.D. (1996). Flexible smoothing with B-splines and penalties (with comments and rejoinder). Statistical Science, 11(2): 89-121.

Examples

```
# Gaussian series
library(JOPS)
# Simulate data
m = 20
sig =.1
n = 5
set.seed(2021)
x = seq(0, 1, length = m)
Y = matrix(0, m, n)
a = rnorm(n)
for (j in 1:n){
    adj = a[j] + rnorm(m) * sig
    Y[, j] = sin(2 * x) + adj}
y = as.vector(Y)
x = rep(x, n)
unit = rep(1:n, each = m)
fit1 = psRandInt(y, x, unit, family ='gaussian', se = 2)
names(fit1)
plot(fit1)
# Poisson series
library(JOPS)
# Simulate data
m = 20
sig =.1
n = 5
set.seed(2021)
x = seq(0, 1, length = m)
Y = matrix(0, m, n)
a = rnorm(n)
for (j in 1:n) {
 v = sin(2 * x) + a[j]
 Y[, j] = rpois(m, exp(v + 2))
}
y = as.vector(Y)
x = rep(x, n)
unit = rep(1:n, each = m)
fit2 = psRandInt(y, x, unit, family ='poisson', se = 2)
names(fit2)
plot(fit2)
# Binomial series
library(JOPS)
# Simulate data
m = 20
n = 2
set.seed(2021)
```

```
x = seq(0, 1, length = m)
Y = matrix(0, m, n)
a = rnorm(n)
mbinomial = rep(1, m)
for (j in 1:n){
  v = sin(2 * x) + a[j]
 pii = 10/(10+exp(v + 2))
 Y[, j] = rbinom(m, mbinomial, pii)}
y = as.vector(Y)
x = rep(x, n)
mbinomialv = rep(mbinomial, n)
unit = rep(1:n, each = m)
fit3 = psRandInt(y, x, unit, family= 'binomial', m_binomial = mbinomialv, se = 2)
names(fit3)
plot(fit3)
```

`plot.sim2Dpsr` *Plotting function for* `sim_2Dpsr`

Description

Plotting function for single-index signal regression with tensor product P-splines (using `sim_psr` with `class simpsr`).

Usage

```
## S3 method for class 'sim2Dpsr'
plot(x, ..., xlab = " ", ylab = " ", Resol = 100)
```

Arguments

`x`	the P-spline object, usually from `sim_psr`.
`...`	other parameters.
`xlab`	label for the x-axis, e.g. "my x" (quotes required).
`ylab`	label for the y-axis, e.g. "my y" (quotes required).
`Resol`	resolution for plotting, default `Resol = 100`.

Value

`Plot`	a two panel plot, one for the estimated P-spline signal coefficent surface, and another for the estimated (unkown) P-spline smooth link function.

Author(s)

Paul Eilers, Brian Marx, and Bin Li

References

Marx, B.D., P.H.C. Eilers, and B. Li (2011). Multidimensional single-index signal regression. Chemometrics and Intelligent Laboratory Systems. 109: 120-130.

Eilers, P.H.C., B. Li, B.D. Marx (2009). Multivariate calibration with single-index signal regression, *Chemometrics and Intellegent Laboratory Systems*, 96(2), 196-202.

Eilers, P.H.C. and Marx, B.D. (2021). *Practical Smoothing, The Joys of P-splines.* Cambridge University Press.

Examples

```
library(fields)
library(JOPS)
library(JOPSplus)
# Get the data
x0<- Sugar$X
x0 <- x0 - apply(x0, 1, mean) # center Signal
y <- as.vector(Sugar$y[, 3]) # Response is Ash

# Inputs for two-dimensional signal regression
nsegs <- c(7, 37, 5)
pords <- c(3, 3, 2)
bdegs = c(3, 3, 3)
lambdas <- c(8850, 420, 0.1) # Can be found via svcm
M1_index <- rev(c(340, 325, 305, 290, 255, 240, 230))
M2_index <- seq(from = 275, to = 560, by = .5)
p1 <- length(M1_index)
p2 <- length(M2_index)
max_iter <- 100
# Single-index 2D model
fit <- sim_2Dpsr(y, x0, p1, p2, M1_index, M2_index, nsegs, bdegs, lambdas, pords,
             max_iter, M_type = "unfolded")

plot(fit, xlab = " ", ylab = " ")
predict(fit, M_pred = x0[1:3,], M_type = 'unfolded')
```

`predict.gpls` *Predict function for* `gpls`

Description

Prediction function which returns both linear predictor and inverse link predictions for an arbitrary matrix of signals (using `gpls` with `class gpls`).

Usage

```
## S3 method for class 'gpls'
predict(object, ..., X_pred, type = "mu")
```

Arguments

object	an object using gpls.
...	other parameters.
X_pred	a matrix of q arbitrary signal vectors of dimension q by p1 for desired prediction.
type	the mean value type = "mu" (default) or linear predictor type = "eta".

Value

pred	the estimated mean (inverse link function) (default) or the linear predictor prediction with type = "eta", at signals in matrix X_pred.

Author(s)

Brian Marx

References

Marx, B.D. (1996). Iteratively reweighted partial least squares estimation for generalized linear regression. Technometrics 38(4): 374-381.

Examples

```
library(JOPS)
library(JOPSplus)
# Get the data
library(fds)
data(nirc)
iindex <- nirc$x
X <- nirc$y
sel <- 50:650 # 1200 <= x & x<= 2400
X <- X[sel, ]
iindex <- iindex[sel]
dX <- diff(X)
diindex <- iindex[-1]
y <- as.vector(labc[1, 1:40]) # percent fat
oout <- 23
dX <- t(dX[, -oout])
y <- y[-oout]
fit1 <- gpls(y, dX, components=2, family = "gaussian", link = "identity", X_pred = dX)
plot(fit1, xlab = "Coefficient Index", ylab = "gpls coefficients")
title(main = "GPLS coefficient vector")
names(fit1)
predict(fit1, X_pred = dX[1:5,], type ='mu')
```

---

predict.ps2dvcsignal *Predict function for* ps2DVCSignal

Description

Prediction function which returns both linear predictor and inverse link predictions for an arbitrary matrix of signals with their vectors of companion indexing covariates (using ps2DVCSignal with class ps2dvcsignal).

Usage

```
## S3 method for class 'ps2dvcsignal'
predict(object, ..., X_pred, t1_pred, t2_pred, type = "mu")
```

Arguments

object	an object using ps2DVCSignal.
...	other parameters.
X_pred	a matrix of q arbitrary signal vectors of dimension q by p1 for desired prediction.
t1_pred	a q vector for the first varying index variable associated with X_pred.
t2_pred	a q vector for the second varying index variable associated with X_pred.
type	the mean value type = "mu" (default) or linear predictor type = "eta".

Value

pred	the estimated mean (inverse link function) (default) or the linear predictor prediction with type = "eta", at signals in matrix X_pred and covariates in vectors t1_pred and t2_pred.

Author(s)

Paul Eilers and Brian Marx

References

Eilers, P. H. C. and Marx, B. D. (2003). Multivariate calibration with temperature interaction using two-dimensional penalized signal regression. *Chemometrics and Intellegent Laboratory Systems*, 66, 159–174.

Eilers, P.H.C. and Marx, B.D. (2021). *Practical Smoothing, The Joys of P-splines.* Cambridge University Press.

Examples

```
library(JOPS)
library(JOPSplus)
fit1 <- ps2DVCSignal(y = rnorm(30), X = matrix(rnorm(30*100),30, 100), 1:100, t1_var = rnorm(30),
t2_var = rnorm(30), t1_coef_loc= c(0,1), t2_coef_loc=c(-1,0))
plot(fit1, xlab = "Coefficient Index", ylab = "Signal coef.", lty = c(1,2))
names(fit1)
predict(fit1, X_pred = matrix(rnorm(5*100),5,100), t1_pred =
rnorm(5)/3, t2_pred = rnorm(5)/3)
```

predict.sim2Dpsr *Predict function for* sim_2Dpsr

Description

Prediction function which returns single-index inverse link linear predictions at arbitrary signal/images (using sim_2Dpsr with class sim2Dpsr).

Usage

```
## S3 method for class 'sim2Dpsr'
predict(object, ..., M_pred, M_type = "stacked")
```

Arguments

object	an object using sim_2Dpsr.
...	other parameters.
M_pred	a matrix of q arbitrary "stacked" or "unfolded" signal matrices of dimension (q by p1) by p2 or q by (p1 by p2, respectively, for desired prediction (default "unfolded").
M_type	"stacked" (default) or "unfolded".

Value

pred	the estimated (inverse single-index) mean for the signals in X_pred.

Author(s)

Paul Eilers and Brian Marx

References

Marx, B.D., P.H.C. Eilers, and B. Li (2011). Multidimensional single-index signal regression. Chemometrics and Intelligent Laboratory Systems. 109: 120-130.

Eilers, P.H.C., B. Li, B.D. Marx (2009). Multivariate calibration with single-index signal regression, *Chemometrics and Intellegent Laboratory Systems*, 96(2), 196-202.

Eilers, P.H.C. and Marx, B.D. (2021). *Practical Smoothing, The Joys of P-splines.* Cambridge University Press.

Examples

```
library(fields)
library(JOPS)
library(JOPSplus)
# Get the data
x0<- Sugar$X
x0 <- x0 - apply(x0, 1, mean) # center Signal
y <- as.vector(Sugar$y[, 3]) # Response is Ash

# Inputs for two-dimensional signal regression
nsegs <- c(7, 37, 5)
```

```
pords <- c(3, 3, 2)
bdegs = c(3, 3, 3)
lambdas <- c(8850, 420, 0.1) # Can be found via svcm
M1_index <- rev(c(340, 325, 305, 290, 255, 240, 230))
M2_index <- seq(from = 275, to = 560, by = .5)
p1 <- length(M1_index)
p2 <- length(M2_index)
max_iter <- 100
# Single-index 2D model
fit <- sim_2Dpsr(y, x0, p1, p2, M1_index, M2_index, nsegs, bdegs, lambdas, pords,
             max_iter, M_type = "unfolded")

plot(fit, xlab = " ", ylab = " ")
predict(fit, M_pred = x0[1:3,], M_type = 'unfolded')
```

`ps2DVCSignal` *Two-dimensional varying-coefficient penalized signal regression using P-splines.*

Description

`ps2DVCSignal` is used to regress a (glm) response onto a signal such that the signal coefficients can vary over two other covariates `t1_var` and `t2_var`. Anisotripic penalization of triple tensor product B-splines produces a 3D coefficient surface that can be sliced at a (`t1_var`, `t2_var`) location.

@details Support functions needed: `pspline_fitter`, `pspline_2dchecker`, and `bbase`.

@import stats @import JOPS

Usage

```
ps2DVCSignal(
  y,
  X,
  x_index,
  t1_var,
  t2_var,
 Pars = rbind(c(min(x_index), max(x_index), 10, 3, 1, 2), c(min(t1_var), max(t1_var),
   5, 3, 1, 2), c(min(t2_var), max(t2_var), 5, 3, 1, 2)),
  family = "gaussian",
  link = "default",
  m_binomial = 1 + 0 * y,
  wts = 1 + 0 * y,
  r_gamma = 1 + 0 * y,
  X_pred = X,
  t1_pred = t1_var,
  t2_pred = t2_var,
  y_predicted = NULL,
  ridge_adj = 1e-08,
  int = TRUE,
  t1_coef_loc = 0,
  t2_coef_loc = 0,
  Resol = 100
)
```

Arguments

y	a glm response vector of length m, usually continuous, binary/bimomial or counts.
X	a m by p1 Signal matrix of regressors.
x_index	p1-vector for index of Signal (e.g. wavelength).
t1_var	p2-vector with other (indexing) variable in coefficient surface (e.g. temperature, depth, time).
t2_var	p3-vector with other (indexing) variable in coefficient surface (e.g. temperature, depth, time).
Pars	a matrix with 3 rows, each with P-spline parameters: min max nseg bdeg lambda pord, for row and columns of tensor product surface; defaults are min and max for x_index and t_var (resp.), nseg = 10, bdeg =3, lambda = 1, pord = 2.
family	the response distribution, e.g. "gaussian","binomial","poisson","Gamma" distribution; quotes are needed (default "gaussian".
link	the link function, one of "identity", "log", "sqrt", "logit", "probit", "cloglog", "loglog", "reciprocal"); quotes are needed (default "identity".
m_binomial	a vector of binomial trials having length(y). Default is 1 vector for family = "binomial", NULL otherwise.
wts	a m vector of weights (default 1).
r_gamma	a vector of gamma shape parameters. Default is 1 vector for family = "Gamma", NULL otherwise.
X_pred	a matrix of signals with ncol(X) columns for prediction, default is X.
t1_pred	a vector for the VC indexing variable with length nrow(X_pred), default is t1_var.
t2_pred	a vector for the VC indexing variable with length nrow(X_pred), default is t2_var.
y_predicted	a vector for the responses associated with X_pred with length nrow(X_pred) useful for CV when family = "binomial", default is NULL.
ridge_adj	a small ridge penalty tuning parameter to regularize estimation (default 1e-8).
int	intercept set to TRUE or FALSE for intercept term.
t1_coef_loc	vector of t1_var locations for estimated signal coefficient.
t2_coef_loc	vector of t2_var locations for estimated signal coefficient.
Resol	Resolution of estimated signal coefficient vector at (t1_coef_loc, t2_coef_loc).

Value

pcoef	a vector of length (Pars[1,3]+Pars[1,4])*(Pars[2,3]+Pars[2,4]) of estimated P-spline coefficients for tensor surface.
summary_predicted	inverse link prediction vectors, and twice se bands.
dev	the deviance of fit.
eff_df	the approximate effective dimension of fit.
family	the family of the response.
link	the link function.
aic	AIC.

df_resid	approximate df residual.
cv	leave-one-out standard error prediction when family = "gaussian", NULL otherwise.
cv_predicted	standard error prediction for y_predict when family = "gaussian", NULL otherwise.
Pars	design and tuning parameters; see arguments above.
dispersion_parm	estimate of dispersion, Dev/df_resid.
summary_predicted	inverse link prediction vectors, and twice se bands.
eta_predicted	estimated linear predictor of length(y).
press_mu	leave-one-out prediction of mean when family = "gaussian", NULL otherwise.
bin_percent_correct	percent correct classification based on 0.5 cut-off when family = "binomial", NULL otherwise.
Bx	B-spline basis matrix of dimension p1 by n1, along x_index.
By	B-spline basis matrix of dimension m by n2, along t1_var.
Bz	B-spline basis matrix of dimension m by n3, along t2_var.
Q	Modified tensor basis (m by (n1*n2)) for VC signal regression.
yint	the estimated y-intercept (when int = TRUE.)
int	a logical variable related to use of y-intercept in model.
Ahat_m	matrix (nrow(Ahat_m)=Resol) of estimated signal coefficient vectors at (t1_coef_loc, t2_coef_loc.)
t1_coef_loc	t1_var coordinate for signal coefficent (default 0).
t2_coef_loc	t2_var coordinate for signal coefficent (default 0).
Resol	resolution of estimated signal coefficient vector (default 100).

Author(s)

Paul Eilers and Brian Marx

References

Eilers, P.H.C. and Marx, B.D. (2003). Multivariate calibration with temperature interaction using two-dimensional penalized signal regression. *Chemometrics and Intellegent Laboratory Systems*, 66, 159–174.

Eilers, P.H.C. and Marx, B.D. (2021). *Practical Smoothing, The Joys of P-splines.* Cambridge University Press.

Examples

```
library(JOPS)
library(JOPSplus)
fit1 <- ps2DVCSignal(y = rnorm(30), X = matrix(rnorm(30*100),30, 100), 1:100, t1_var = rnorm(30),
t2_var = rnorm(30), t1_coef_loc= c(0,1), t2_coef_loc=c(-1,0))
plot(fit1, xlab = "Coefficient Index", ylab = "Signal coef.", lty = c(1,2))
names(fit1)
predict(fit1, X_pred = matrix(rnorm(5*100),5,100), t1_pred =
rnorm(5)/3, t2_pred = rnorm(5)/3)
```

psDensity *Univarite density estimation using (Poisson) P-splines.*

Description

psDensity is used to estimate the smooth density of a vector x using Poisson P-splines on (finely) constructed histograms.

Usage

```
psDensity(
  x,
  xl = min(x),
  xr = max(x),
  m = 10 * log(length(x)),
  nseg = 40,
  bdeg = 3,
  pord = 3,
  lambda = 10
)
```

Arguments

x	the covariate vector to which a density estimate is computed.
xl	the number for the min along x (default is min(x)).
xr	the number for the max along x (default is max(x)).
m	the number of bins for the (finely) constructed histogram (default m = 10 * log(length(data))).
nseg	the number of evenly spaced segments between xl and xr (default 40).
bdeg	the number of the degree of the basis, usually 1, 2, or 3 (defalult).
pord	the number of the order of the difference penalty, usually 1, 2 (default), or 3.
lambda	the (positive) number for the tuning parameter for the penalty (default 10).

Value

dev	Deviance of fit.
effdim	effective dimension of fit.
aic	AIC.
nseg	the number of B-spline segments.
bdeg	the degree of the B-spline basis.
pord	the order of the difference penalty.
lambda	the positive tuning parameter.
family	the family of the response ("Poisson").
link	the link function used ("log").
xgrid	gridded x values, useful for plotting.
ygrid	gridded Poisson linear predictor values, useful for plotting.
dgrid	gridded fitted (inverse link) density values, useful for plotting.

Author(s)

Paul Eilers and Brian Marx

References

Eilers, P.H.C. and Marx, B.D. (2021). *Practical Smoothing, The Joys of P-splines.* Cambridge University Press.

Eilers, P.H.C., Marx, B.D., and Durban, M. (2015). Twenty years of P-splines, *SORT*, 39(2): 149-186.

Eilers, P.H.C. and Marx, B.D. (1996). Flexible smoothing with B-splines and

Examples

```
fitN = psDensity(rnorm(200), lambda = 1000)
plot(fitN, ylab= 'pdf', xlab = 'z-score')

#@examples
library(JOPS)
library(boot)
Year <- coal$date
xl <- min(Year)
xr <- max(Year)

# Poisson smoothing
nseg <- 50
bdeg <- 3
# Note: it is important to consider reasonalble xl, xr values.
fit1 <- psDensity(Year, xl = 1845, xr = 1965, m = 40, nseg, bdeg, pord = 3, lambda = 100)
plot(fit1, xlab = "Year", ylab = "Count")
```

---

psRandInt *Shared P-spline curves with random individual offsets for Gaussian, Poisson, binomial responses; Penalty updated in mixed model style*

Description

psRandInt computes shared P-spline curves with random individual intercept offsets for Gaussian, Poisson, or binomial responses, updated penalties in mixed model style.

Usage

```
psRandInt(
  y,
  x,
  unit = c(1:length(y)),
  xl = min(x),
  xr = max(x),
  nseg = 10,
  bdeg = 3,
  pord = 3,
  family = "gaussian",
```

```
  m_binomial = 0 * y + 1,
  se = 0
)
```

Arguments

y	the response, which can be gaussian (default), Poisson, or binomial.
x	the covariate vector to which a smooth estimate is computed (e.g. time).
unit	the unique ID corresponding to the subject series, unequal length allowed.
xl	the number for the min along x (default is min(x)), useful for extrapolation.
xr	the number for the max along x (default is max(x)), useful for extrapolation.
nseg	the number of evenly spaced segments between xl and xr (default 40).
bdeg	the number of the degree of the basis, usually 1, 2, or 3 (default).
pord	the number of the order of the difference penalty, usually 1, 2 (default), or 3.
family	the response distribution, e.g."gaussian","binomial","poisson" distribution; quotes are needed. Default is "gaussian".
m_binomial	number of binomial trials (default=1).
se	the non-negative value associated with the number of standard errors used in se bands; e.g. 1.96 or 2, default = 0 (suppresses se output).

Value

dev	deviance of fit.
effdim	effective dimension of fit.
aic	AIC.
nseg	the number of B-spline segments.
bdeg	the degree of the B-spline basis.
pord	the order of the difference penalty.
lambda_B	the (HFS) positive tuning parameter for smooth.
lambda_int	the (HFS) positive tuning parameter for intercepts.
family	the family of the response.
link	the link function used.
xgrid	gridded x values, useful for plotting.
ygrid	gridded linear predictor values, useful for plotting.
mugrid	gridded fitted (inverse link) values, useful for plotting.
se_eta	gridded standard errors for the linear predictor.
se_bands_int	CIs with se * standard error for the intercepts.
nu	the number of unique units (subjects).
mgrid	the resolution of the gridded plot.
xl	the number for the min along x (default is min(x)), useful for extrapolation.
xr	the number for the max along x (default is max(x)), useful for extrapolation.
x	the covariate vector to which a smooth estimate is computed (e.g. time).
y	the response, which can be gaussian (default), Poisson, or binomial.

unit	the unique ID corresponding to the subject series.
se	the non-negative value associated with the number of standard errors used in se bands; e.g. 1.96 or 2, default = 0 (suppresses se output)
.	
disp_parm	the GLM dispersion parameter, MSE for gaussian, set to 1 for Poisson, binomial

Author(s)

Paul Eilers and Brian Marx

References

Eilers, P.H.C. and Marx, B.D. (2021). *Practical Smoothing, The Joys of P-splines.* Cambridge University Press.

Eilers, P.H.C., Marx, B.D., and Durban, M. (2015). Twenty years of P-splines, *SORT*, 39(2): 149-186.

Eilers, P.H.C. and Marx, B.D. (1996). Flexible smoothing with B-splines and penalties (with comments and rejoinder). Statistical Science, 11(2): 89-121.

Examples

```
# Gaussian series
library(JOPS)
# Simulate data
m = 20
sig =.1
n = 5
set.seed(2021)
x = seq(0, 1, length = m)
Y = matrix(0, m, n)
a = rnorm(n)
for (j in 1:n){
    adj = a[j] + rnorm(m) * sig
    Y[, j] = sin(2 * x) + adj}
y = as.vector(Y)
x = rep(x, n)
unit = rep(1:n, each = m)
fit1 = psRandInt(y, x, unit, family ='gaussian', se = 2)
names(fit1)
plot(fit1)
# Poisson series
library(JOPS)
# Simulate data
m = 20
sig =.1
n = 5
set.seed(2021)
x = seq(0, 1, length = m)
Y = matrix(0, m, n)
a = rnorm(n)
for (j in 1:n) {
 v = sin(2 * x) + a[j]
 Y[, j] = rpois(m, exp(v + 2))
}
```

```
y = as.vector(Y)
x = rep(x, n)
unit = rep(1:n, each = m)
fit2 = psRandInt(y, x, unit, family ='poisson', se = 2)
names(fit2)
plot(fit2)
# Binomial series
library(JOPS)
# Simulate data
m = 20
n = 2
set.seed(2021)
x = seq(0, 1, length = m)
Y = matrix(0, m, n)
a = rnorm(n)
mbinomial = rep(1, m)
for (j in 1:n){
  v = sin(2 * x) + a[j]
 pii = 10/(10+exp(v + 2))
 Y[, j] = rbinom(m, mbinomial, pii)}
y = as.vector(Y)
x = rep(x, n)
mbinomialv = rep(mbinomial, n)
unit = rep(1:n, each = m)
fit3 = psRandInt(y, x, unit, family= 'binomial', m_binomial = mbinomialv, se = 2)
names(fit3)
plot(fit3)
```

`sim_2Dpsr` *Single-Index for 2D signal regression using P-splines*

Description

`sim_2Dpsr` is a single-index signal regression model that estimates both the signal coefficients surface and the unknown link function using P-splines.

Usage

```
sim_2Dpsr(
  y,
  M,
  p1,
  p2,
  M1_index = c(1:p1),
  M2_index = c(1:p2),
  nsegs = rep(10, 3),
  bdegs = rep(3, 3),
  lambdas = rep(10, 3),
  pords = rep(2, 3),
  max_iter = 100,
  M_type = "stacked"
)
```

Arguments

y	a response vector of length m, usually continuous.
M	The signal/image regressors, which are either "stacked" or "unfolded", with dimensions (m * p1) by p2 (i.e. m stacked matrices each of p1 by p2) or with dimensions m by (p1 * p2) (i.e. regressor matrix with m regressor rows, each with column length p1 * p2), respectively.
p1	the row dimension of the image.
p2	the column dimension of the image.
M1_index	an index of length p1 for rows of regressor matrix (default is a simple sequence).
M2_index	an index of length p2 for columns of regressor matrix (default is a simple sequence).
nsegs	a vector of length 3 containing the number of evenly spaced segments between min and max, for each the coefficient surface (row/col) and the (unknown) link function, resp. (default c(10,10,10)).
bdegs	a vector of length 3 containing the degree of B-splines, for the coefficient surface (row/col) and the (unknown) link function, resp. (default cubic or c(3,3,3)).
lambdas	a vector of length 3 containing the positive tuning parameters, for each the coefficient surface (row/col) and the (unknown) link function, resp. (default c(1,1,1)).
pords	a vector of length 3 containing the difference penalty order, for each the coefficient surface (row/col) and the (unknown) link function, resp. (defaultc(2,2,2)).
max_iter	a scalar for the maximum number of iterations (default 100).
M_type	"stacked" (default) (signal as matrix) or "unfolded" (signal as vector).

Value

y	the response vector of length m.
alpha	the P-spline coefficient vector of length (nsegs[1]+bdeg[1])*nsegs[2]+bdeg[2].
iter	the number of iterations used for the single-index fit.
yint	the estimated y-intercept for the single-index model.
B	the B-spline matrix built along the signal index, using nsegs[1], used for the coefficient vector.
Q	the effective regressors from the psVCSignal portion of the single-index fit with dimension m by length(alpha).
nsegs	a vector of length 3 containing the number of evenly spaced segments between min and max, for each the coefficient surface (row/col) and the link function, resp.
bdegs	a vector of length 2 containing the degree of B-splines, for each the coefficient vector and the link function, resp.
lambdas	a vector of length 3 containing the positive tuning parameters, for each the coefficient surface (row/col) and the link function, resp.
pords	a vector of length 3 containing the difference penalty order, for each the coefficient surface (row/col) and the link function, resp.
eta	the estimated linear predictor for the single-index fit.

cv	the leave-one-out cross-validation statistic or the standard error of prediction for the single-index fit.
delta_alpha	change measure in signal-coefficent parameters at convervence.
M1_index	the index of length p1 for columns of signal matrix.
M2_index	the index of length p2 for columns of signal matrix.
f_fit	the psNormal object, fitting link function f(eta).
f_eta	the predicted values of the link function estimated with f_fit or estimated f(eta), at x = eta.

Author(s)

Paul Eilers, Brian Marx, and Bin Li

References

Marx, B.D., P.H.C. Eilers, and B. Li (2011). Multidimensional single-index signal regression. Chemometrics and Intelligent Laboratory Systems. 109: 120-130.

Eilers, P.H.C., B. Li, B.D. Marx (2009). Multivariate calibration with single-index signal regression, *Chemometrics and Intellegent Laboratory Systems*, 96(2), 196-202.

Eilers, P.H.C. and Marx, B.D. (2021). *Practical Smoothing, The Joys of P-splines.* Cambridge University Press.

Examples

```
library(fields)
library(JOPS)
library(JOPSplus)
# Get the data
x0<- Sugar$X
x0 <- x0 - apply(x0, 1, mean) # center Signal
y <- as.vector(Sugar$y[, 3]) # Response is Ash

# Inputs for two-dimensional signal regression
nsegs <- c(7, 37, 5)
pords <- c(3, 3, 2)
bdegs = c(3, 3, 3)
lambdas <- c(8850, 420, 0.1) # Can be found via svcm
M1_index <- rev(c(340, 325, 305, 290, 255, 240, 230))
M2_index <- seq(from = 275, to = 560, by = .5)
p1 <- length(M1_index)
p2 <- length(M2_index)
max_iter <- 100
# Single-index 2D model
fit <- sim_2Dpsr(y, x0, p1, p2, M1_index, M2_index, nsegs, bdegs, lambdas, pords,
             max_iter, M_type = "unfolded")

plot(fit, xlab = " ", ylab = " ")
predict(fit, M_pred = x0[1:3,], M_type = 'unfolded')
```

Index