

# Package: pblm (via r-universe)

May 16, 2026

**Type** Package

**Title** Bivariate Additive Marginal Logistic Regression via Maximum Penalized Likelihood Estimation

**Version** 0.1-8

**Date** 2017-04-2

**Author** Marco Enea, with contributions by Mikis Stasinopoulos and Robert Rigby

**Maintainer** Marco Enea <marco.enea@unipa.it>

**Depends** R (>= 3.2.2), Matrix, lattice, splines, MASS

**Imports** methods

**Description** Bivariate additive categorical regression for moderate-to-small size datasets. Under a multinomial scheme, it is possible to fit bivariate models when the two responses are nominal, ordinal or mixed nominal/ordinal. Partial proportional odds models with (non-)uniform association structure can be fitted with the possibility to specify several logit types and parametrizations for the marginals and the association, including the Dale's model. The association structure can also be smoothed using penalty terms of polynomial type. P-splines are used in the additive part of the model. Common methods such as summary, residuals and predict are available.

**License** GPL (>= 2)

**LazyData** TRUE

**NeedsCompilation** no

**URL** <http://github.com/enea/pblm>

**BugReports** <http://github.com/enea/pblm/issues>

**Repository** <https://marcoenea.r-universe.dev>

**Date/Publication** 2020-02-06 10:32:11 UTC

**RemoteUrl** <https://github.com/marcoenea/pblm>

**RemoteRef** HEAD

**RemoteSha** 1e77d719ceee6a734f1e1cb9e627a174ac3c81b9

**RemoteSubdir** pblm

## Contents

pblm-package . . . . .	2
bms . . . . .	3
multicolumn . . . . .	3
pb . . . . .	4
pblm . . . . .	5
pblm.control . . . . .	12
pblm.penalty . . . . .	13
pblm.prop . . . . .	16
plot.pblm . . . . .	18
summary.pblm . . . . .	19
ulcer . . . . .	21

**Index** 22

---

pblm-package	<i>Bivariate Additive Marginal Logistic Regression via Maximum Penalized Likelihood Estimation</i>
--------------	--

---

## Description

Fitting bivariate additive marginal logistic regression models for categorical responses.

## Details

Package: pblm  
 Type: Package  
 Version: 0.1-8  
 Date: 2017-04-02  
 License: GPL (>= 2)

It is possible to fit partial proportional odds models and to specify several parametrizations for the marginals and the association, including the Dale's model. P-splines can be used to smooth covariates. The association structure can also be smoothed by polynomial surfaces by using penalty terms.

## Author(s)

Marco Enea, with contributions by Mikis Stasinopoulos and Robert Rigby

Maintainer: Marco Enea <marco.enea@libero.it>

---

bms *A British male sample on occupational status.*

---

**Description**

These data were analyzed by Goodman (1979) and concern the cross-classification of a sample of fathers and their sons according to the occupational status.

**Usage**

```
data(bms)
```

**Format**

A data frame with 49 observations and 3 variables.

fathers fathers' occupational status. A factor with levels from 1 to 7.

sons sons' occupational status. A factor with levels from 1 to 7.

freq a vector of integers representing the number of people cross-classified according to the occupational status of fathers and sons

**Source**

Goodman, L. A. (1979). Simple models for the analysis of cross-classifications having ordered categories. *Journal of the American Statistical Association*, **74**(367), 537-552.

**Examples**

```
data(bms)
xtabs(freq ~ fathers + sons, data=bms)
```

---

multicolumn *transforming bivariate data in a multi-column format*

---

**Description**

This function transforms a grouped two-column response data frame into a multi-column one, another data format accepted by `pb1m`

**Usage**

```
multicolumn(formula, data)
```

**Arguments**

formula	a two-side formula with counts in the left side.
data	a data frame with two categorical responses, covariates (if any) and a count variable.

**Value**

A data frame with as many responses as the number cells from the underlying response table and covariates (if any).

**Author(s)**

Marco Enea

**Examples**

```
#NOT RUN
data(ulcer)
multicolumn(freq~medication+pain+operation,data=ulcer)
```

---

pb *Specify a Penalised B-Spline Fit in a pblm Formula*

---

**Description**

Both pb and pbs are adaptations of function pb and ps from the [gamlss](#) package, respectively, to specify penalized B-spline.

**Usage**

```
pb(x, df = NULL, lambda = NULL, control = pb.control(...), ...)
pb.control(inter = 20, degree = 3, order = 2, quantiles = FALSE, ...)
pbs(x, df = 3, lambda = NULL, ps.intervals = 20, degree = 3, order = 3)
```

**Arguments**

x	the univariate predictor.
df	the desired equivalent number of degrees of freedom (trace of the smoother matrix minus two for the constant and linear fit).
lambda	the smoothing parameter.
control	setting the control parameters
ps.intervals	the number of break points in the x-axis.
inter	the number of break points (knots) in the x-axis.
degree	the degree of the piecewise polynomials.
order	the required difference in the vector of coefficients.
quantiles	if TRUE the quantile values of x are used to determine the knots.
...	for extra arguments.

**Details**

Basically, pb is a reduced-functionality version of the original one specified in `gamlss` with no performance iteration methods (i.e. by method specification) implemented. The only method implemented minimizes the GAIC by internal `optim` calls.

**Author(s)**

Marco Enea, based on the original versions of the corresponding functions contained in the `gamlss` package by Mikis Stasinopoulos and Bob Rigby.

**Examples**

```
#NOT RUN

# an artificial data set:
set.seed(1234)
da <- expand.grid("Y1"=1:3,"Y2"=1:3,"fat1"=0:9,"fat2"=0:1)
da$x1 <- seq(-5,5,l=180)
da$x2 <- rnorm(180)

da$Freq <- sample(5:30,180,replace=TRUE)

m1 <- pblm(fo1=cbind(Y1,Y2) ~ pbs(x1) + fat2,
          fo2=~pb(x1) + x2,
          fo12=~pb(x1) + x2, data=da, weights=Freq)

par(mfrow=c(2,3))
plot(m1)
```

---

pblm

*Bivariate Additive Regression for Categorical Responses*

---

**Description**

This function allows to fit bivariate additive marginal logistic regression models for nominal, ordinal or mixed nominal/ordinal responses.

**Usage**

```
pblm(fo1=NULL, fo2=NULL, fo12=NULL, RC.fo=NULL, data, weights=NULL,
     type="gg", verbose=FALSE, contrasts=NULL, start=NULL, x=FALSE,
     center=FALSE, scale=FALSE, plackett=NULL, ncat1=NULL, ncat2=NULL,
     fit=TRUE, proportional=pblm.prop(...), penalty=pblm.penalty(...),
     control=pblm.control(...), ...)
```

**Arguments**

fo1	a two hand-side formula for the logit(s) of the first response. Depending on the data structure, the left-hand side can be a two-column or a multi-column response vector. In the latter case, argument ncat1 must be specified. See Examples.
fo2	a one hand-side formula for the logit(s) of the second response. If omitted, it will be assumed to be equal to fo1.
fo12	a one hand-side formula for the log-odds ratio(s) of the association between the two responses. If omitted, it will be assumed to be equal to fo1.
RC.fo	a Row/Column type formula specifying the association structure. See Details.
data	a data frame.
weights	An optional vector containing the observed frequencies.
type	a two-length string specifying the type of logits to use in the model fit. See Details.
verbose	logical. Should information about convergence be printed during model estimation?
contrasts	the Row/Column contrasts to be used in RC.fo. See argument contrasts.arg in <a href="#">model.matrix.default</a> .
x	logical. Should the model matrix used in the fitting process be returned as component of the fitted model?
center	logical. Should the covariates be centered with respect their mean before the fit?
scale	logical. Should the covariates be scaled with respect their standard deviation before the fit?
start	an optional vector of starting values for the coefficients of the non-additive part.
plackett	logical. Should a Plackett-based formula be used for the inversion $\eta \rightarrow \pi$ . Actually, this is allowed (and it is the default) for binary or ordered responses only. The more general method uses the Newton-Raphson inversion algorithm described in Glonek and McCullagh (1995).
ncat1	an integer indicating the number of levels of the first response. Mandatory the data is in multicolumn format. See example below.
ncat2	an integer indicating the number of levels of the second response.
fit	logical. If TRUE (default), the model will be estimated, otherwise only some objects created prior to estimation will be returned.
control	this sets the control parameters for the Fisher-scoring and the inner Newton-Raphson and backfitting algorithms. The default setting is specified by the <a href="#">pblm.control</a> function.
proportional	this sets a list of logical vectors specifying which explanatory variables depend on the categories of the responses. The default setting is specified by the <a href="#">pblm.prop</a> function
penalty	this sets the penalty terms and smoothing parameters for a non-parametric "vertical" smoothing across the categories of the responses. The default setting is specified by the <a href="#">pblm.penalty</a> function.
...	further arguments.

## Details

It is possible to fit partial proportional odds models and specify several association structures like the Goodman's (1979) model (interactions are allowed though these are of linear type), the Dale's (1986) model and their additive version (Bustami *et al.*, 2001), Enea *et al.* (2014). Furthermore, the association structure can also be smoothed by using penalty terms of polynomial type as those considered in Enea and Attanasio (2015). That allows to enlarge the range of possible parametrizations of the association structure as an alternative to the Dale's Row-Column parametrization. Further details on the penalty terms specified through `pblm.penalty` can also be found in Enea and Lovison (2014).

The algorithm is based on the bivariate version of the model by Glonek and McCullagh (1995), that is by using the *multivariate logit transform*

$$C' \log(M\pi) = \eta = X\beta.$$

Once `fo1` has been specified, if `fo2` and `fo12` are left unspecified, these are assumed to be equal to `fo1`. By default, the function fits a proportional odds model for ordered responses, in which only the marginal and the association intercepts are assumed to be category-dependent (Glonek and McCullagh, 1995).

Model formulae using a Row-Column type parametrization, like in the Goodman or the Dale models, need to be specified by using `RC.fo`. The right-hand side of such formula only recognizes `Row` and `Col` to specify row and/or column effects. Covariates must be specified separately by using `fo12`. See Examples.

The logits implemented are *local*; *global*; *continuation*; *reverse continuation*; (Colombi and Forcina, 2001) and *basic*. By using argument `type`, several log-odds ratios can be specified, among the permutations of the local-local type (`type="ll"`), local-global ("`lg`"),..., basic-basic (`type="bb"`). Furthermore, if the responses are binary, setting `type="ss"` will correspond to classical logit  $\pi = \log P(Y = 1)/P(Y = 0)$  for both responses, while other specifications will produce  $\log P(Y = 0)/P(Y = 1)$ .

The vector of the starting values must be set in the following order: intercepts of the first logit; covariates of the first logit; intercepts of the second logit; covariates of the second logit; association intercepts; association covariates.

For what concerning the additive part, p-slines can be fitted by using `pb` and/or `pbs`, which are adaptations (reduced versions) of `pb` and `ps` from the `gamlss` package, respectively.

## Value

A list of components from the model fit. Some of these could be redundant and removed in a future version of the package. When `fit=TRUE` the returned components are:

<code>coef</code>	a named column vectors of coefficients.
<code>n</code>	the total number of observations.

m	the number of observed configurations of the responses given covariates (if any). It corresponds to the number of rows of the dataset.
p	fitted probability matrix given the observations.
Y	the weighted matrix of the responses in multinomial format.
x	if requested, the model matrix.
xx1, xx2, xx12	vectors, for internal use.
ynames	vector with the names of the responses.
tol	the accuracy reached at the convergence.
llp	the penalized log-likelihood value at the convergence.
ll	the log-likelihood value at the convergence.
devp	the penalized deviance value at the convergence.
dev	the deviance value at the convergence.
IM	the estimated Information Matrix at the convergence.
Imp	the estimated penalized Information Matrix at the convergence. Its inverse is used to calculate the standard error of estimates.
convergence	logical indicating whether convergence criteria were satisfied.
iter	the number of iterations performed in the model fitting.
maxB	the number of smoothers present in the equation with the maximum number of smoothers. This also represents the number of outer iterations of the backfitting algorithm.
ncat1, ncat2	the number of levels for the two responses.
ncat	this is simply $ncat1 * ncat2$ .
weights	the prior weights, that is the weights initially supplied, a vector of 1s if none were.
P	a $p \times p$ penalty matrix, where $p$ is the number of parameters estimated (including all the intercepts), concerning the penalty term specified (if any), but excluding the additive part (if any). If no penalty terms are specified, this will be a matrix of zeros.
gaic.m	the <i>penalty</i> per parameter of the generalized AIC initially provided. It is 2 by default.
lam1, lam2, \cr lam3, lam4	vectors of smoothing parameters initially supplied with the specification of a penalty term. If not, these are NULL.
opt	the object returned by <code>optim</code> if automatic selection of smoothing parameters has been set. NULL otherwise.
etasmooth	the matrix of predicted values for the additive part, NULL otherwise.
eta	the $n \times (ncat-1)$ matrix of predicted values on the observed data.
fsmooth	a list of objects initially created by the smoothers (if any), NULL otherwise.
one.smooth	for internal use.

<code>df.fix</code>	the degrees of freedom of the parametric part of the model. Note that if a penalty term is used by specifying <code>penalty</code> , the corresponding degrees of freedoms will be counted in the parametric part of the model.
<code>df.fix.vect</code>	for internal use.
<code>df.smooth</code>	the degrees of freedom of the additive part of the model (if any), otherwise zero.
<code>w.res</code>	a $n \times (ncat-1)$ matrix of working residuals.
<code>W2</code>	a $m$ -length list of matrices of working weights.
<code>z</code>	a $(m*ncat-1)$ -length working vector of responses.
<code>any.smoother</code>	logical indicating whether smoothers have been used in the model fit.
<code>Bmat</code>	list of bases of smoothers used in the model fit (if any), NULL otherwise.
<code>wh.eq</code>	list of logical vectors indicating which terms in the linear predictor are smoothers.
<code>wh.eq2</code>	list of logical vectors indicating which terms in the linear predictor are smoothers. For internal use.
<code>PBWB</code>	list of numerical matrices if smoothers are used, NULL otherwise. For internal use.
<code>BWB</code>	list of numerical matrices if smoothers are used, NULL otherwise. For internal use.
<code>etalist</code>	list of numerical vectors if smoothers are used. For internal use.
<code>spec.name</code>	list of numerical vectors if smoothers are used. For internal use.
<code>beta.smooth</code>	list of estimated coefficients for the smoothers (if any), NULL otherwise.
<code>n.smoothers</code>	the number of smoothers used.
<code>PPmat</code>	list of penalty matrices for the smoothers multiplied by the smoothing parameters. NULL if smoothers are not used.
<code>pn1.type</code>	the type of penalty used.
<code>xnames</code>	list of vectors with the names of the covariates.
<code>prop.smooth</code>	for internal use.
<code>GAIC</code>	the value of the generalized AIC from the fitted model for the specified value of <code>gaic.m</code> in <code>pblm.control</code> .
<code>ta</code>	the underlying observed contingency table of the responses, marginally to the covariates.
<code>set0</code>	the parameter setting from <code>pblm.control</code> as specified by the user.
<code>fo.list</code>	list of formulas used.
<code>center</code>	logical indicating whether argument if the covariates were centered
<code>scale</code>	logical indicating whether argument if the covariates were scaled
<code>type</code>	the type of logit used.
<code>plackett</code>	logical indicating whether the plackett inversion formula was used.
<code>RC.fo</code>	the Row-Column formula as specified by the user.
<code>contrasts</code>	the Row-Column formula contrasts as specified by the user.
<code>acc2</code>	tolerance to be used for the estimation as specified by the user.
<code>maxit</code>	maximum number of Fisher-scoring iterations as specified by the user.

**Warning**

Please be sure that the results you get are really what you are "expecting" when using uncommon specifications of argument type, mainly those involving an s logit type and its combinations with other logits. A part from the binary case, many of those have been not checked yet in the current version of the package.

The estimation of category-dependent p-splines, as outlined in Enea et al. (2014), is not allowed (work in progress).

**Note**

Please note that specifying a formula with interaction terms in RC. fo corresponds to a model with association structure of the type  $\alpha + \beta_r + \gamma_c + \delta_{rc}$ . In the current version of the package, non linear interaction terms of the type  $\alpha + \beta_r + \gamma_c + \delta_{1r}\delta_{2c}$ , as considered for example in Lapp et al. (1998), are not implemented here.

Furthermore, unlikely from the Dale's parameterization, pblm does not put a minus sign to covariates.

**Author(s)**

Marco Enea <marco.enea@unipa.it> with contribution by Mikis Stasinopoulos and Bob Rigby

**References**

Bustami, R., Lesaffre, E., Molenberghs, G., Loos, R., Danckaerts, M. and Vlietinck, R. 2001. Modelling bivariate ordinal responses smoothly with examples from ophthalmology and genetics. *Statistics in Medicine*, **20**, 1825-1842.

Colombi, R. and Forcina, A. 2001. Marginal regression models for the analysis of positive association of ordinal response variables. *Biometrika*, **88**(4), 1007-1019.

Dale, J. R. (1986) Global Cross-Ratio Models for Bivariate, Discrete, Ordered Responses. *Biometrics*, **42**(4), 909-917.

Enea, M. and Attanasio, M. 2015. A model for bivariate data with application to the analysis of university students success. *Journal of Applied Statistics*, <http://dx.doi.org/10.1080/02664763.2014.998407>

Enea, M. and Lovison, G. 2014. A Penalized penalized approach to the bivariate logistic regression model for the association between ordinal responses. arXiv:1407.1751 [math.ST]

Enea, M., Stasinopoulos, M., Rigby, R., and Plaia, A. 2014. The pblm package: semiparametric regression for bivariate categorical responses in R. In Thomas Kneib, Fabian Sobotka, Jan Fahrenholz, Henriette Irmer (Eds.) *Proceeding of the 29th International Workshop of Statistical Modelling*, Volume 2, 47-50.

Glonek, G. F. V. and McCullagh, P. (1995) Multivariate logistic models. *Journal of the Royal Statistical Society, Series B*, **57**, 533-546.

Goodman, L. A. (1979). Simple models for the analysis of cross-classifications having ordered categories. *Journal of the American Statistical Association*, **74**(367), 537-552.

Lapp, K., Molenberghs, G., and Lesaffre, E. (1998) Models for the association between ordinal variables. *Computational Statistics and Data Analysis*, **27**, 387-411.

### See Also

[summary.pblm](#)

### Examples

```
## Example 1
#The Dale's model
data(ulcer)
m1 <- pblm(fo1=cbind(pain,medication)~1, fo12=~I(operation=="vh"), RC.fo=~Col,
          data=ulcer, weights=ulcer$freq, contrasts=list(Col="contr.SAS"))

#compare with Dale (1986), Table 3
summary(m1)
deviance(m1)

#the same data in another format
dat <- multicolumn(freq~medication+pain+operation,data=ulcer)
dat

fo <- as.formula(paste(attributes(dat)$"resp", "~1", sep=""))

m1bis <- pblm(fo1=fo, fo12=~I(operation=="vh"), RC.fo=~Col, verbose=TRUE,
            data=dat, ncat1=3, contrasts=list(Col="contr.SAS"))

deviance(m1bis)

## Example 2
#NOT RUN
# an artificial example:
set.seed(10)
da <- expand.grid("Y1"=1:3,"Y2"=1:3,"fat1"=0:4,"fat2"=0:1)
da$Freq <- sample(0:20,3*3*5*2,replace=TRUE)
da$x1 <- rnorm(90)

#the bivariate additive proportional-odds model
m2 <- pblm(fo1=cbind(Y1,Y2) ~ fat1 + pb(x1), data=da, weights=da$Freq)
par(mfrow=c(1,3))
plot(m2)
```

pblm.control

*Auxiliary for controlling the algorithm in a pblm model***Description**

This is an auxiliary function for controlling the algorithm in a pblm model.

**Usage**

```
pblm.control(maxit = 30, maxit2 = 200, acc = 1e-07, acc2 = 1e-06,
             zero.adj = 1e-06, l = NULL, restore.l = FALSE,
             min.step.l = 1e-04, auto.select = FALSE, gaic.m = 2,
             rss.tol = 1e-06, max.backfitting = 10, pgtol.df = 0.01,
             factr.df = 1e+07, lmm.df = 5, parscale.df = 1,
             max.gaic.iter = 500, pgtol.gaic = 1e-05, grad.tol = 1e-07,
             factr.gaic = 1e+07, lmm.gaic = 5, parscale = 1,
             conv.crit = c("dev", "pdev"))
```

**Arguments**

maxit	maximum number of Fisher-scoring iterations.
maxit2	maximum number of Newton-Raphson iterations for the inversion $\eta \rightarrow \pi$ .
acc	tolerance to be used for the estimation.
acc2	tolerance to be used for the inversion $\eta \rightarrow \pi$ .
zero.adj	adjustment factor for zeros in the probability vector $\pi$ .
l	numerical, ranged in (0,1], representing the initial value of step length. By default l=1.
restore.l	logical, should the step length be restored to its initial value after each iteration? This is an experimental option and may be changed in the future.
min.step.l	numerical, minimum value fixed for the step length.
auto.select	logical, should the smoothing parameters be estimated by GAIC minimization? If TRUE The optimization will be performed numerically by using <code>optim</code> .
gaic.m	the "penalty" per parameter of the generalized AIC. By default it is 2, corresponding to the classical AIC.
rss.tol	tolerance for the residual sum of squares used in the backfitting algorithm.
max.backfitting	maximum number of backfitting iterations.
pgtol.df	tolerance to be used in order to get an amount of smoothing corresponding to the fixed degrees of freedom for the additive part. See argument <code>pgtol</code> from <code>optim</code> .
factr.df	numerical. For degrees-of-freedom optimization in the additive part. See argument <code>factr</code> from <code>optim</code> .
lmm.df	integer. For degrees-of-freedom optimization in the additive part. See argument <code>lmm</code> from <code>optim</code> .

parscale.df	A vector of scaling parameters for vector lambda when optimizing lambda for fixed degrees of freedom. See argument <code>parscale</code> from <a href="#">optim</a> .
max.gaic.iter	integer. Maximum number of iterations for automatic model optimization. See argument <code>maxit</code> from <a href="#">optim</a> .
pgtol.gaic	numerical. Tolerance to be used for automatic selection of smoothing parameters. See argument <code>pgtol</code> from <a href="#">optim</a> .
grad.tol	numerical. Tolerance to be used when inverting the gradient matrix.
factr.gaic	numerical. For automatic selection of smoothing parameters. See argument <code>factr</code> from <a href="#">optim</a> .
lmm.gaic	integer. For automatic selection of smoothing parameters. See argument <code>lmm</code> from <a href="#">optim</a> .
parscale	A vector of scaling parameters for vector lambda for automatic model optimization. See argument <code>parscale</code> from <a href="#">optim</a> .
conv.crit	Convergence criterion for model estimation. The default is "dev", corresponding to log-likelihood maximization. Alternatively, "pdev" is concerned with maximum penalized log-likelihood.

**Value**

A list with the same arguments of the function, unless unlikely specified by the user.

**Author(s)**

Marco Enea

**See Also**

[pblm](#)

---

pblm.penalty                      *Auxiliary for specifying penalty terms in a pblm model*

---

**Description**

This is an auxiliary function for specifying penalty terms in a pblm model.

**Usage**

```
pblm.penalty(pnl.type=c("none", "ARC1", "ARC2", "ridge", "lasso", "lassoV", "equal"),
             lam1=NULL, lam2=NULL, lam3=NULL, lam4=NULL,
             s1=NULL, s2=NULL, s3=NULL, s4=NULL, min.lam.fix=0.1,
             constraints=FALSE, lamv1=1e10, lamv2=1e10)
```

**Arguments**

<code>pn1.type</code>	The type of penalty term to be used. By default "none" of them is used. See Details.
<code>lam1, lam2</code>	vectors of smoothing parameters for the marginals. By default they are zero. It is common to all the penalty terms implemented.
<code>lam3</code>	vector of smoothing parameters for the association. It is common to all the penalty terms implemented. If "ARC2" is selected, it will act on differences of by-row adjacent parameters.
<code>lam4</code>	vector smoothing parameters for the association. Specific for the "ARC2" penalty, for which the differences of by-column adjacent parameters are penalized
<code>s1, s2, s3, s4</code>	the orders of the difference operator. Specific for the "ARC2" penalty.
<code>min.lam.fix</code>	the minimum value for any penalty parameters in order to consider any lamdas smaller than this value as fixed. See details.
<code>constraints</code>	Should inequality constraints be applied to the marginal probabilities? This is done through a penalty term and intended for ordered responses.
<code>lamv1, lamv2</code>	penalty parameters to be applied to both the marginal probabilities in order to mimicking inequality constraints.

**Details**

Some penalty terms implemented in `pblm` are described in Enea and Lovison (2014) and Enea and Attanasio (2015).

Just one penalty per model is allowed.

Penalty "ARC1" acts on first order differences of category-adjacent parameters. By increasing the smoothing parameters, the resulting marginal and/or association parameters, will tend to be equal among the categories. When the underlying contingency table cross-classifying the responses contains zero cells, This penalty may be useful to stabilize the estimates, for example to get a more "regular" association structure.

Penalty "ARC2" generalizes in a certain sense "ARC1", since it acts on high order differences of Adjacent Row and/or Column parameters, but it is mainly used for ordered responses. For high smoothing values it constraints the marginal parameters to lie onto a polynomial curve, and/or the association structure to lie onto a polynomial surface. The degrees of the marginal polynomials are determined by  $s1-1$ , for the first marginal, and  $s2-1$  for the second. The degree of the association polynomial surface is determined by  $s3+s4-2$ , in which  $s3-1$  is the by-row polynomial degree and  $s4-1$  the by-column one.

Penalty "ridge" constraints the regression parameters towards zero (horizontal penalty), so providing equal estimates for high penalty values.

The current implementation of the "lasso" and "lassoV" penalty terms is to be considered provisional and needs to be better checked.

Penalty "lasso" is acts similarly to "ridge" and it is based on absolute values of the regression parameters.

Penalty "lassoV" vertically penalizes the absolute value of differences of adjacent row and column parameters. It is similar to ARC1. By increasing the smoothing parameter, the resulting marginal and/or association regression parameters, will tend to be equal among the response categories. This

Penalty "equal" constraints the marginal equations to be equal, so providing equal estimates. This could be useful, for example, in eyes or twins studies. The tuning parameter to be specified for this penalty is lam1.

Furthermore, if global logits are specified, inequality constraints on marginal predictors are mimicked by using penalty terms. Argument lamv1 is the penalty parameter for the marginal predictor of the first response, lamv2 is that for the second one.

Argument min.lam.fix is useful when an automatic selection of penalty parameters is desired for certain lambdas only. The remaining lambdas can be either 0 or less than min.lam.fix, and excluded from the automatic selection. Fixing some lambdas to assume values in  $[0, \text{min.lam.fix}]$  may be useful, for example, for parameter space regularization.

### Value

A list with the same arguments of the function, unless unlikely specified by the user.

### Author(s)

Marco Enea <marco.enea@unipa.it>

### References

Enea, M. and Attanasio, M. 2015. A model for bivariate data with application to the analysis of university students success. Journal of Applied Statistics, <http://dx.doi.org/10.1080/02664763.2014.998407>

Enea, M. and Lovison, G. 2014. A Penalized penalized approach to the bivariate logistic regression model for the association between ordinal responses. arXiv:1407.1751 [math.ST]

### See Also

[pblm](#)

### Examples

```
#Example 1
# A British male sample on occupational status.
data(bms)

# A third degree polynomial surface with equally-spaced integer scores
```

```

m1 <- pblm(fo1=cbind(fathers,sons)~1, data=bms, weights=bms$freq,
           penalty=pblm.penalty(pnl.type="ARC2",lam3=c(1e7), lam4=c(1e7),
                                s3=c(4), s4=c(4)))

require(lattice)
g <- expand.grid("sons"=1:6,"fathers"=1:6)
g$logGOR <- m1$coef[13:48]

wireframe(logGOR ~ sons*fathers, data = g, zlim=c(min(g$logGOR-1),max(g$logGOR+1)),
          scales = list(arrows = FALSE), screen = list(z = -130, x = -70),
          col.regions="magenta")

#Example 2

# an artificial data frame with two binary responses and two factors
set.seed(12)
da <- expand.grid("Y1"=0:1,"Y2"=0:1,"fat1"=0:1,"fat2"=0:1)
da$Freq <- sample(0:20,2*2*2*2,replace=TRUE)

# A quasi-independence model obtained by strongly penalizing the association intercept
# through a ridge-type penalty term
m3 <- pblm(fo1=cbind(Y1,Y2) ~ fat1 + fat2,
          fo12=~ 1,
          data=da, weights=da$Freq, type="ss",
          proportional=pblm.prop(prop12=c(TRUE)),
          penalty=pblm.penalty(pnl.type="ridge",lam3=1e12))
summary(m3)

# notice that the last coefficient is not exactly zero
coef(m3)

m3.1 <- glm(Y1 ~ fat1 + fat2, data=da, weights=Freq, family=binomial)
m3.2 <- glm(Y2 ~ fat1 + fat2, data=da, weights=Freq, family=binomial)

all.equal(logLik(m3), logLik(m3.1)[1]+logLik(m3.2)[1])

```

**Description**

This is an auxiliary function which allows to specify partially proportional odds for one (or both) the marginals and with the association parameters which can depend (or not) on the categories of the responses. It simply returns a list with its arguments.



plot.pblm

*Plotting smoothers for a pblm object***Description**

Plotting smoothers for a pblm object

**Usage**

```
## S3 method for class 'pblm'
plot(x, which.eq = 1:3, which.var = 1:x$maxNpred, add.bands = TRUE,
      col.bands = "lightblue", col.line = "black", type = "l",
      dashed.bands = FALSE, pause = FALSE, ylim, xlim, ylab, xlab,
      main, ...)
```

**Arguments**

x	Object of class pblm.
which.var	variable index for the smoothers indicating the rank of the smoother as this appear in the model formula.
which.eq	equation index for the smoothers.
add.bands	logical, should the confidence bands for the smoother be added to the graph?
col.bands	sets the color for the smoother confidence bands.
col.line	sets the color for the smoother line.
type	graphical parameter.
dashed.bands	logical. If TRUE (as well as add.bands) dashed (instead of colour filled) confidence bands will be drawn.
pause	logical. If TRUE the user will be asked to press a button to plot each graph.
ylim	graphical parameter.
xlim	graphical parameter.
ylab	graphical parameter.
xlab	graphical parameter.
main	graphical parameter.
...	further arguments.

**Author(s)**

Marco Enea

**See Also**

[pb](#), [pbs](#)

**Examples**

```

#NOT RUN
# an artificial data set:
set.seed(10)
da <- expand.grid("Y1"=1:3,"Y2"=1:3,"fat1"=0:4,"fat2"=0:1)
da$Freq <- sample(0:20,3*3*5*2,replace=TRUE)
da$x1 <- rnorm(90)
#the bivariate additive proportional-odds model
m7 <- pblm(fo1=cbind(Y1,Y2) ~ factor(fat1) + pb(x1), data=da, weights=da$Freq)
par(mfrow=c(1,3))
plot(m7,which.eq=1,main="marginal equation 1")
plot(m7,which.eq=2,main="marginal equation 2")
plot(m7,which.eq=3,main="association")

```

summary.pblm

*Summarizing methods for bivariate additive logistic regression***Description**

Summarizing methods and functions for objects of class pblm.

**Usage**

```

## S3 method for class 'pblm'
summary(object,...)

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

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

## S3 method for class 'pblm'
AIC(object,...,k=2)

## S3 method for class 'pblm'
logLik(object, penalized=FALSE,...)

## S3 method for class 'pblm'
vcov(object,...)

## S3 method for class 'pblm'
coef(object, digits = max(3, getOption("digits") - 3), ...)

## S3 method for class 'pblm'
coefficients(object, digits = max(3, getOption("digits") - 3), ...)

```

```

## S3 method for class 'pblm'
residuals(object, type = c("working", "pearson"),...)

## S3 method for class 'pblm'
resid(object, type = c("working", "pearson"),...)

## S3 method for class 'pblm'
fitted(object,...)

## S3 method for class 'pblm'
predict(object, newdata, type=c("link","response","terms","count"),
        se.fit=FALSE, digits= max(6, getOption("digits") - 3),...)

## S3 method for class 'pblm'
deviance(object, penalized=FALSE,...)

edf.pblm(object, which.var=1, which.eq=1)

se.smooth.pblm(object)

```

### Arguments

object	an object of class pblm.
digits	controls the number of digits printed in the output.
x	an object to be printed produced by pblm or summary.pblm.
k	numeric, the penalty per parameter to be used; by default $k = 2$ , that is the classical AIC.
penalized	logical indicating whether the value of the penalized log-likelihood is required
which.var	variable index indicating the position of the smoother as this appears in the model formula.
which.eq	equation index for the smoothers.
newdata	if provided this represents a data.frame with new values of model covariates.
type	the type of desired residuals or prediction.
se.fit	logical. Do you want prediction standard errors to be returned? Currently allowed only for type="link" when newdata is not specified.
...	further arguments.

### Author(s)

Marco Enea

### See Also

[pblm](#), [pb](#)

**Examples**

```
#NOT RUN
# an artificial data set:
set.seed(10)
da <- expand.grid("Y1"=1:3,"Y2"=1:3,"fat1"=0:4,"fat2"=0:1)
da$Freq <- sample(1:20,3*3*5*2,replace=TRUE)
da$x1 <- rnorm(90)
#the bivariate additive proportional-odds model
m7 <- pblm(fo1=cbind(Y1,Y2) ~ fat1 + pb(x1), data=da, weights=da$Freq)
summary(m7)
par(mfrow=c(1,3))
plot(m7,which.eq=1,xlab="x1",main="marginal equation 1")
plot(m7,which.eq=2,xlab="x1",main="marginal equation 2")
plot(m7,which.eq=3,xlab="x1",main="association")
```

ulcer

*The ulcer data***Description**

Data analyzed by Dale (1986)

**Usage**

```
data(ulcer)
```

**Format**

A data frame with 48 observations and 4 variables.

medication medication requirements. A factor with levels never; seldom; occasionally; and regularly.

pain patients' post operative pain level. A factor with levels none, slight and moderate.

operation a factor representing the type of operation with levels: vagotomy drainage procedure (vp); vagotomy and distal antrectomy (va); vagotomy and hemigastrectomy (vh); and resection alone (ra).

freq a numeric vector representing the number of patients classified for the corresponding levels of pain, medication and operation

**Source**

Dale, J. R. (1986) Global Cross-Ratio Models for Bivariate, Discrete, Ordered Responses. *Biometrics*, **42**(4), 909-917.

**Examples**

```
data(ulcer)
```

# Index

## \* datasets

bms, 3  
ulcer, 21

## \* models

multicolumn, 3  
pb, 4  
pblm, 5  
pblm.control, 12  
pblm.penalty, 13  
pblm.prop, 16  
plot.pblm, 18  
summary.pblm, 19

## \* multivariate

multicolumn, 3  
pb, 4  
pblm, 5  
pblm.control, 12  
pblm.penalty, 13  
pblm.prop, 16  
plot.pblm, 18  
summary.pblm, 19

## \* nonparametric

multicolumn, 3  
pb, 4  
pblm, 5  
pblm.control, 12  
pblm.penalty, 13  
pblm.prop, 16  
plot.pblm, 18  
summary.pblm, 19

## \* regression

multicolumn, 3  
pb, 4  
pblm, 5  
pblm.control, 12  
pblm.penalty, 13  
pblm.prop, 16  
plot.pblm, 18  
summary.pblm, 19

## \* smooth

multicolumn, 3  
pb, 4  
pblm, 5  
pblm.control, 12  
pblm.penalty, 13  
pblm.prop, 16  
plot.pblm, 18  
summary.pblm, 19

AIC.pblm (summary.pblm), 19

bms, 3

chisq.test.pblm (summary.pblm), 19  
coef.pblm (summary.pblm), 19  
coefficients.pblm (summary.pblm), 19

deviance.pblm (summary.pblm), 19

edf.pblm (summary.pblm), 19

fitted.pblm (summary.pblm), 19

gamlss, 4

logLik.pblm (summary.pblm), 19

model.matrix.default, 6  
multicolumn, 3

optim, 5, 8, 12, 13

pb, 4, 4, 7, 18, 20  
pblm, 5, 13, 15, 20  
pblm-package, 2  
pblm.control, 6, 9, 12  
pblm.penalty, 6, 7, 13  
pblm.prop, 6, 16  
pbs, 7, 18  
pbs (pb), 4  
plot.pblm, 18

`predict.pblm(summary.pblm)`, 19  
`print.pblm(summary.pblm)`, 19  
`print.summary.pblm(summary.pblm)`, 19  
`ps`, 4, 7

`resid.pblm(summary.pblm)`, 19  
`residuals.pblm(summary.pblm)`, 19  
`Rsq.pblm(summary.pblm)`, 19

`se.smooth.pblm(summary.pblm)`, 19  
`summary.pblm`, 11, 19

`ulcer`, 21

`vcov.pblm(summary.pblm)`, 19