

# Package ‘gpls’

May 1, 2024

**Title** Classification using generalized partial least squares

**Version** 1.76.0

**Author** Beiying Ding

**Description** Classification using generalized partial least squares for two-group and multi-group (more than 2 group) classification.

**Maintainer** Bioconductor Package Maintainer  
<maintainer@bioconductor.org>

**Imports** stats

**Suggests** MASS

**License** Artistic-2.0

**biocViews** Classification, Microarray, Regression

**git\_url** <https://git.bioconductor.org/packages/gpls>

**git\_branch** RELEASE\_3\_19

**git\_last\_commit** c48f123

**git\_last\_commit\_date** 2024-04-30

**Repository** Bioconductor 3.19

**Date/Publication** 2024-04-30

## Contents

gpls1a . . . . .	2
gpls1a.cv.error . . . . .	3
gpls1a.logit.all . . . . .	4
gpls1a.mlogit . . . . .	5
gpls1a.mlogit.cv.error . . . . .	7
gpls1a.train.test.error . . . . .	8
gpls . . . . .	9
predict.gpls . . . . .	11

<b>Index</b>	<b>13</b>
--------------	-----------

glpls1a

*Fit IRWPLS and IRWPLSF model***Description**

Fit Iteratively ReWeighted Least Squares (IRWPLS) with an option of Firth's bias reduction procedure (IRWPLSF) for two-group classification

**Usage**

```
glpls1a(X, y, K.prov = NULL, eps = 0.001, lmax = 100, b.ini = NULL,
        denom.eps = 1e-20, family = "binomial", link = NULL, br = TRUE)
```

**Arguments**

X	n by p design matrix (with no intercept term)
y	response vector 0 or 1
K.prov	number of PLS components, default is the rank of X
eps	tolerance for convergence
lmax	maximum number of iteration allowed
b.ini	initial value of regression coefficients
denom.eps	small quantity to guarantee nonzero denominator in deciding convergence
family	glm family, binomial is the only relevant one here
link	link function, logit is the only one practically implemented now
br	TRUE if Firth's bias reduction procedure is used

**Value**

coefficients	regression coefficients
convergence	whether convergence is achieved
niter	total number of iterations
bias.reduction	whether Firth's procedure is used
loading.matrix	the matrix of loadings

**Author(s)**

Beiyang Ding, Robert Gentleman

**References**

- Ding, B.Y. and Gentleman, R. (2003) *Classification using generalized partial least squares*.
- Marx, B.D (1996) Iteratively reweighted partial least squares estimation for generalized linear regression. *Technometrics* 38(4): 374-381.

**See Also**

[glpls1a.mlogit](#), [glpls1a.logit.all](#), [glpls1a.train.test.error](#), [glpls1a.cv.error](#), [glpls1a.mlogit.cv.error](#)

**Examples**

```
x <- matrix(rnorm(20),ncol=2)
y <- sample(0:1,10,TRUE)
## no bias reduction
glpls1a(x,y,br=FALSE)

## no bias reduction and 1 PLS component
glpls1a(x,y,K.prov=1,br=FALSE)

## bias reduction
glpls1a(x,y,br=TRUE)
```

---

glpls1a.cv.error	<i>Leave-one-out cross-validation error using IRWPLS and IRWPLSF model</i>
------------------	--

---

**Description**

Leave-one-out cross-validation training set classification error for fitting IRWPLS or IRWPLSF model for two group classification

**Usage**

```
glpls1a.cv.error(train.X,train.y, K.prov=NULL,eps=1e-3,lmax=100,family="binomial",link="logit",br=T)
```

**Arguments**

train.X	n by p design matrix (with no intercept term) for training set
train.y	response vector (0 or 1) for training set
K.prov	number of PLS components, default is the rank of train.X
eps	tolerance for convergence
lmax	maximum number of iteration allowed
family	glm family, binomial is the only relevant one here
link	link function, logit is the only one practically implemented now
br	TRUE if Firth's bias reduction procedure is used

**Value**

error	LOOCV training error
error.obs	the misclassified error observation indices

**Author(s)**

Beiyong Ding, Robert Gentleman

**References**

- Ding, B.Y. and Gentleman, R. (2003) *Classification using generalized partial least squares*.
- Marx, B.D (1996) Iteratively reweighted partial least squares estimation for generalized linear regression. *Technometrics* 38(4): 374-381.

**See Also**

[glpls1a.train.test.error](#), [glpls1a.mlogit.cv.error](#), [glpls1a](#), [glpls1a.mlogit](#), [glpls1a.logit.all](#)

**Examples**

```
x <- matrix(rnorm(20),ncol=2)
y <- sample(0:1,10,TRUE)

## no bias reduction
glpls1a.cv.error(x,y,br=FALSE)
## bias reduction and 1 PLS component
glpls1a.cv.error(x,y,K.prov=1, br=TRUE)
```

---

glpls1a.logit.all	<i>Fit MIRWPLS and MIRWPLSF model separately for logits</i>
-------------------	---

---

**Description**

Apply Iteratively ReWeighted Least Squares (MIRWPLS) with an option of Firth's bias reduction procedure (MIRWPLSF) for multi-group (say C+1 classes) classification by fitting logit models for all C classes vs baseline class separately.

**Usage**

```
glpls1a.logit.all(X, y, K.prov = NULL, eps = 0.001, lmax = 100, b.ini = NULL, denom.eps = 1e-20, family =
```

**Arguments**

X	n by p design matrix (with no intercept term)
y	response vector with class labels 1 to C+1 for C+1 group classification, baseline class should be 1
K.prov	number of PLS components
eps	tolerance for convergence
lmax	maximum number of iteration allowed
b.ini	initial value of regression coefficients
denom.eps	small quantity to guarantee nonzero denominator in deciding convergence

family	glm family, binomial (i.e. multinomial here) is the only relevant one here
link	link function, logit is the only one practically implemented now
br	TRUE if Firth's bias reduction procedure is used

Value

coefficients	regression coefficient matrix
--------------	-------------------------------

Author(s)

Beiyong Ding, Robert Gentleman

References

- Ding, B.Y. and Gentleman, R. (2003) *Classification using generalized partial least squares*.
- Marx, B.D (1996) Iteratively reweighted partial least squares estimation for generalized linear regression. *Technometrics* 38(4): 374-381.

See Also

[glpls1a.mlogit](#),[glpls1a](#),[glpls1a.mlogit.cv.error](#),[glpls1a.train.test.error](#),[glpls1a.cv.error](#)

Examples

```
x <- matrix(rnorm(20),ncol=2)
y <- sample(1:3,10,TRUE)
## no bias reduction
glpls1a.logit.all(x,y,br=FALSE)
## bias reduction
glpls1a.logit.all(x,y,br=TRUE)
```

---

glpls1a.mlogit	<i>Fit MIRWPLS and MIRWPLSF model</i>
----------------	---------------------------------------

---

Description

Fit multi-logit Iteratively ReWeighted Least Squares (MIRWPLS) with an option of Firth's bias reduction procedure (MIRWPLSF) for multi-group classification

Usage

```
glpls1a.mlogit(x, y, K.prov = NULL, eps = 0.001, lmax = 100, b.ini = NULL, denom.eps = 1e-20, family = "b
```

**Arguments**

x	n by p design matrix (with intercept term)
y	response vector with class labels 1 to C+1 for C+1 group classification, baseline class should be 1
K.prov	number of PLS components
eps	tolerance for convergence
lmax	maximum number of iteration allowed
b.ini	initial value of regression coefficients
denom.eps	small quantity to guarantee nonzero denominator in deciding convergence
family	glm family, binomial (i.e. multinomial here) is the only relevant one here
link	link function, logit is the only one practically implemented now
br	TRUE if Firth's bias reduction procedure is used

**Value**

coefficients	regression coefficient matrix
convergence	whether convergence is achieved
niter	total number of iterations
bias.reduction	whether Firth's procedure is used

**Author(s)**

Beiyang Ding, Robert Gentleman

**References**

- Ding, B.Y. and Gentleman, R. (2003) *Classification using generalized partial least squares*.
- Marx, B.D (1996) Iteratively reweighted partial least squares estimation for generalized linear regression. *Technometrics* 38(4): 374-381.

**See Also**

[glpls1a](#), [glpls1a.mlogit.cv.error](#), [glpls1a.train.test.error](#), [glpls1a.cv.error](#)

**Examples**

```
x <- matrix(rnorm(20),ncol=2)
y <- sample(1:3,10,TRUE)
## no bias reduction and 1 PLS component
glpls1a.mlogit(cbind(rep(1,10),x),y,K.prov=1,br=FALSE)
## bias reduction
glpls1a.mlogit(cbind(rep(1,10),x),y,br=TRUE)
```

---

`glpls1a.mlogit.cv.error`*Leave-one-out cross-validation error using MIRWPLS and MIRW-PLSF model*

---

**Description**

Leave-one-out cross-validation training set error for fitting MIRWPLS or MIRWPLSF model for multi-group classification

**Usage**

```
glpls1a.mlogit.cv.error(train.X, train.y, K.prov = NULL, eps = 0.001, lmax = 100, mlogit = T, br = T)
```

**Arguments**

<code>train.X</code>	n by p design matrix (with no intercept term) for training set
<code>train.y</code>	response vector with class labels 1 to C+1 for C+1 group classification, baseline class should be 1
<code>K.prov</code>	number of PLS components
<code>eps</code>	tolerance for convergence
<code>lmax</code>	maximum number of iteration allowed
<code>mlogit</code>	if TRUE use the multinomial logit model, otherwise fit all C-1 logistic models (vs baseline class 1) separately
<code>br</code>	TRUE if Firth's bias reduction procedure is used

**Value**

<code>error</code>	LOOCV training error
<code>error.obs</code>	the misclassified error observation indices

**Author(s)**

Beiyong Ding, Robert Gentleman

**References**

- Ding, B.Y. and Gentleman, R. (2003) *Classification using generalized partial least squares*.
- Marx, B.D (1996) Iteratively reweighted partial least squares estimation for generalized linear regression. *Technometrics* 38(4): 374-381.

**See Also**

[glpls1a.cv.error](#), [glpls1a.train.test.error](#), [glpls1a](#), [glpls1a.mlogit](#), [glpls1a.logit.all](#)

**Examples**

```
x <- matrix(rnorm(20),ncol=2)
y <- sample(1:3,10,TRUE)

## no bias reduction
glpls1a.mlogit.cv.error(x,y,br=FALSE)
glpls1a.mlogit.cv.error(x,y,mlogit=FALSE,br=FALSE)
## bias reduction
glpls1a.mlogit.cv.error(x,y,br=TRUE)
glpls1a.mlogit.cv.error(x,y,mlogit=FALSE,br=TRUE)
```

---

```
glpls1a.train.test.error
```

*out-of-sample test set error using IRWPLS and IRWPLSF model*

---

**Description**

Out-of-sample test set error for fitting IRWPLS or IRWPLSF model on the training set for two-group classification

**Usage**

```
glpls1a.train.test.error(train.X,train.y,test.X,test.y,K.prov=NULL,eps=1e-3,lmax=100,family="binomi
```

**Arguments**

train.X	n by p design matrix (with no intercept term) for training set
train.y	response vector (0 or 1) for training set
test.X	transpose of the design matrix (with no intercept term) for test set
test.y	response vector (0 or 1) for test set
K.prov	number of PLS components, default is the rank of train.X
eps	tolerance for convergence
lmax	maximum number of iteration allowed
family	glm family, binomial is the only relevant one here
link	link function, logit is the only one practically implemented now
br	TRUE if Firth's bias reduction procedure is used

**Value**

error	out-of-sample test error
error.obs	the misclassified error observation indices
predict.test	the predicted probabilities for test set



**Author(s)**

Beiyang Ding, Robert Gentleman

**References**

- Ding, B.Y. and Gentleman, R. (2003) *Classification using generalized partial least squares*.
- Marx, B.D (1996) Iteratively reweighted partial least squares estimation for generalized linear regression. *Technometrics* 38(4): 374-381.

**See Also**

[glpls1a.cv.error](#), [glpls1a.mlogit.cv.error](#), [glpls1a](#), [glpls1a.mlogit](#), [glpls1a.logit.all](#)

**Examples**

```
x <- matrix(rnorm(20),ncol=2)
y <- sample(0:1,10,TRUE)
x1 <- matrix(rnorm(10),ncol=2)
y1 <- sample(0:1,5,TRUE)

## no bias reduction
glpls1a.train.test.error(x,y,x1,y1,br=FALSE)
## bias reduction
glpls1a.train.test.error(x,y,x1,y1,br=TRUE)
```

---

gpls

*A function to fit Generalized partial least squares models.*

---

**Description**

Partial least squares is a commonly used dimension reduction technique. The paradigm can be extended to include generalized linear models in several different ways. The code in this function uses the extension proposed by Ding and Gentleman, 2004.

**Usage**

```
gpls(x, ...)
```

## Default S3 method:

```
gpls(x, y, K.prov=NULL, eps=1e-3, lmax=100, b.ini=NULL,
      denom.eps=1e-20, family="binomial", link=NULL, br=TRUE, ...)
```

## S3 method for class 'formula'

```
gpls(formula, data, contrasts=NULL, K.prov=NULL,
      eps=1e-3, lmax=100, b.ini=NULL, denom.eps=1e-20, family="binomial",
      link=NULL, br=TRUE, ...)
```

**Arguments**

<code>x</code>	The matrix of covariates.
<code>formula</code>	A formula of the form ' $y \sim x_1 + x_2 + \dots$ ', where $y$ is the response and the other terms are covariates.
<code>y</code>	The vector of responses
<code>data</code>	A <code>data.frame</code> to resolve the formula, if used
<code>K.prov</code>	number of PLS components, default is the rank of $X$
<code>eps</code>	tolerance for convergence
<code>lmax</code>	maximum number of iteration allowed
<code>b.ini</code>	initial value of regression coefficients
<code>denom.eps</code>	small quantity to guarantee nonzero denominator in deciding convergence
<code>family</code>	glm family, binomial is the only relevant one here
<code>link</code>	link function, <code>logit</code> is the only one practically implemented now
<code>br</code>	TRUE if Firth's bias reduction procedure is used
<code>...</code>	Additional arguments.
<code>contrasts</code>	an optional list. See the <code>contrasts.arg</code> of <code>model.matrix.default</code> .

**Details**

This is a different interface to the functionality provided by [glpls1a](#). The interface is intended to be simpler to use and more consistent with other machine learning code in R.

The technology is intended to deal with two class problems where there are more predictors than cases. If a response variable ( $y$ ) is used that has more than two levels the behavior may be unusual.

**Value**

An object of class `gpls` with the following components:

<code>coefficients</code>	The estimated coefficients.
<code>convergence</code>	A boolean indicating whether convergence was achieved.
<code>niter</code>	The total number of iterations.
<code>bias.reduction</code>	A boolean indicating whether Firth's procedure was used.
<code>family</code>	The family argument that was passed in.
<code>link</code>	The link argument that was passed in.
<code>terms</code>	The constructed terms object.
<code>call</code>	The call
<code>levs</code>	The factor levels for prediction.

**Author(s)**

B. Ding and R. Gentleman

## References

- Ding, B.Y. and Gentleman, R. (2003) *Classification using generalized partial least squares*.
- Marx, B.D (1996) Iteratively reweighted partial least squares estimation for generalized linear regression. *Technometrics* 38(4): 374-381.

## See Also

[glpls1a](#)

## Examples

```
library(MASS)
m1 = gpls(type~., data=Pima.tr, K=3)
```

---

predict.gpls	<i>A prediction method for gpls.</i>
--------------	--------------------------------------

---

## Description

A simple prediction method for gpls objects.

## Usage

```
## S3 method for class 'gpls'
predict(object, newdata, ...)
```

## Arguments

object	A gpls object, typically obtained from a call to <a href="#">gpls</a>
newdata	New data, for which predictions are desired.
...	Other arguments to be passed on

## Details

The prediction method is straight forward. The estimated coefficients from object are used, together with the new data to produce predicted values. These are then split, according to whether the predicted values is larger or smaller than 0.5 and predictions returned.

The code is similar to that in [glpls1a.train.test.error](#) except that in that function both the test and train matrices are centered and scaled (the covariates) by the same values (those from the test data set).

## Value

A list of length two:

class	The predicted classes; one for each row of newdata.
predicted	The estimated predictors.

**Author(s)**

B. Ding and R. Gentleman

**See Also**

[gpls](#)

**Examples**

```
example(gpls)  
p1 = predict(m1)
```

# Index

## \* **classif**

gpls, [9](#)

predict.gpls, [11](#)

## \* **regression**

glpls1a, [2](#)

glpls1a.cv.error, [3](#)

glpls1a.logit.all, [4](#)

glpls1a.mlogit, [5](#)

glpls1a.mlogit.cv.error, [7](#)

glpls1a.train.test.error, [8](#)

glpls1a, [2](#), [4–7](#), [9–11](#)

glpls1a.cv.error, [3](#), [3](#), [5–7](#), [9](#)

glpls1a.logit.all, [3](#), [4](#), [4](#), [7](#), [9](#)

glpls1a.mlogit, [3–5](#), [5](#), [7](#), [9](#)

glpls1a.mlogit.cv.error, [3–6](#), [7](#), [9](#)

glpls1a.train.test.error, [3–7](#), [8](#), [11](#)

gpls, [9](#), [11](#), [12](#)

predict, gpls-method (predict.gpls), [11](#)

predict.gpls, [11](#)

print.gpls (gpls), [9](#)