Package ‘BiplotML’

October 12, 2022

Title Biplots Estimation with Algorithms ML
Version 1.1.0
Date 2022-04-22
Maintainer Jose Giovany Babativa-Marquez <gbabativam@gmail.com>
Depends R (>= 4.1), optimr, optimx, shapes
Description Logistic Biplot is a method that allows representing multivariate binary data on a sub-
space of low dimension, where each individual is represented by a point and each vari-
able as vectors directed through the origin. The orthogonal projection of individu-
als onto these vectors predicts the expected probability that the characteristic occurs. The pack-
age contains new techniques to estimate the model parameters and con-
structs in each case the 'Logistic-Biplot'. References can be found in the help of each procedure.
License MIT + file LICENSE
Encoding UTF-8
LazyData true
RoxygenNote 7.1.2
Suggests testthat, knitr, rmarkdown, RSpectra, dplyr (>= 1.0.0), tidyr
       (>= 1.1.0), ggplot2 (>= 3.3.2), ggrepel, pracma, mvtnorm
URL https://github.com/jgbabativam/BiplotML
BugReports https://github.com/jgbabativam/BiplotML/issues
NeedsCompilation no
Author Jose Giovany Babativa-Marquez [cre, aut]
       (<https://orcid.org/0000-0002-4989-7459>)
Repository CRAN
Date/Publication 2022-04-22 21:20:02 UTC

R topics documented:

bootBLB ................................................................. 2
cv_LogBip ............................................................... 4
fitted_LB ................................................................. 5
Fitting a Binary Logistic Biplot using bootstrap methodology

Description
This function estimates the vector $\mu$, matrix A and matrix B using the optimization algorithm chosen by the user and applies a bootstrap methodology to determine the confidence ellipses.

Usage
bootBLB(
  x,
  k = 2,
  L = 0,
  method = "CG",
  type = 1,
  plot = TRUE,
  sup = TRUE,
  ellipses = FALSE,
  maxit = NULL,
  resamples = 100,
  conf = 0.9,
  col.ind = NULL
)

Arguments
x Binary matrix.
k Dimensions number. By default k = 2.
L Penalization parameter. By default L = 0.
method Method to be used to estimate the parameters. By default method="CG"
type For the conjugate-gradients method. Takes value 1 for the Fletcher–Reeves update, 2 for Polak–Ribiere and 3 for Beale–Sorenson.
plot Plot the Bootstrap Logistic Biplot.
**bootBLB**

**sup**  Boolean, if TRUE, rows that are not selected in each resample are treated as supplementary individuals. See details.

**ellipses**  Draw confidence ellipses. By default is FALSE.

**maxit**  The maximum number of iterations. Defaults to 100 for the gradient methods, and 500 without gradient.

**resamples**  Number of iterations in the bootstrap process. By default 100.

**conf**  Level confidence in the ellipses. By default conf=0.90

**col.ind**  Color for the rows.

**Details**

Fitting when sup=TRUE ... whereas sup=FALSE ...

**Value**

Coordinates of the matrix A and B in resamples and Biplot

**Author(s)**

Giovany Babativa <gbabativam@gmail.com>

**References**


**See Also**

plotBLB, performanceBLB

**Examples**

data("Methylation")
set.seed(02052020)
out.sup <- bootBLB(x = Methylation, ellipses = FALSE)
out <- bootBLB(x = Methylation, sup = FALSE, ellipses = TRUE)
**cv_LogBip**  
*Cross-Validation for logistic biplot*

**Description**

This function runs cross-validation for logistic biplot.

**Usage**

```r
cv_LogBip(
  data,
  k = 0:5,
  K = 7,
  method = "MM",
  type = NULL,
  plot = TRUE,
  maxit = NULL
)
```

**Arguments**

- `data`: Binary matrix.
- `k`: Dimensions to analyze. By default `k = 1:3`.
- `method`: Method to be used to estimate the parameters. By default `method = "MM"`.
- `plot`: Draw the graph. By default `plot = TRUE`.
- `maxit`: The maximum number of iterations. Defaults to 100 for the gradient methods, and 2000 for MM algorithm.

**Value**

Training error and generalization error for a logistic biplot model.

**Author(s)**

Giovany Babativa <gbativam@gmail.com>

**References**


fitted_LB

See Also
LogBip, pred_LB, fitted_LB, simBin

Examples

set.seed(1234)
x <- simBin(n = 100, p = 50, k = 3, D = 0.5, C = 20)
# cross-validation with coordinate descendent MM algorithm
cv_MM <- cv_LogBip(data = x$X, k=0:5, method = "MM", maxit = 1000)

# cross-validation with CG Fletcher-Reeves algorithm
cv.CG <- cv_LogBip(data = x$X, k=0:5, method = "CG", type = 1)

# cross-validation with projection data and block coordinate descending algorithm
cv_PB <- cv_LogBip(data = x$X, k=0:5, method = "PDLB", maxit = 1000)

fitted_LB

Fitted values using Logistic Biplot

Description

Compute the predicted matrix or log-odds for a logistic biplot model

Usage

fitted_LB(object, type = c("link", "response"))

Arguments

object BiplotML object

type the type of fitting required. type = "link" gives output on the logit scale and type = "response" gives output on the probability scale

Value

This function returns the predicted matrix or the log-odds of a binary logistic biplot model.

Author(s)

Giovany Babativa <gbabativam@gmail.com>
Examples

```r
data("Methylation")
LB <- LogBip(Methylation, plot = FALSE)
Theta <- fitted_LB(LB, type = "link")
Pi <- fitted_LB(LB, type = "response")
```

### Description

This function computes the parameters of A and B in Binary Logistic Biplot under algorithm of Descendent Gradient.

### Usage

```r
gradientDesc(
  x,
  k = 2,
  rate = 0.001,
  converg = 0.001,
  max_iter,
  plot = FALSE,
  ...
)
```

### Arguments

- `x`: Binary matrix.
- `k`: Dimensions number. By default `k = 2`.
- `rate`: The value of the rate of descent $\alpha$ in the algorithm of descending gradient. By default $\alpha = 0.001$.
- `converg`: Tolerance limit to achieve convergence. By default `converg = 0.001`.
- `max_iter`: Maximum iterations number.
- `plot`: Plot the Logistic Biplot.
- `...`: other arguments

### Details

We note that the Binary Logistic Biplot is defined as:

$$
\text{logit}(\pi_{ij}) = \log \left( \frac{\pi_{ij}}{1 - \pi_{ij}} \right) = \mu_j + \sum_{s=1}^{k} b_{js}a_{is} = \mu_j + a_{i}^{T}b_{j}
$$
Also, note that the gradient is:
\[ \nabla \ell = \begin{pmatrix} \frac{\partial \ell}{\partial \mu} & \frac{\partial \ell}{\partial A} & \frac{\partial \ell}{\partial B} \end{pmatrix} = (\Pi - X)^T \{(\Pi - X)B, (\Pi - X)^T A\} \]

**Value**

The coefficients of A and B matrix.

**Author(s)**

Giovany Babativa <gbabativam@gmail.com>

**References**


**See Also**

plotBLB, performanceBLB

**Examples**

```r
data(‘Methylation’)  
set.seed(02052020)  
MatGD <- gradientDesc(x = Methylation, k=2, max_iter=10000)  
outGD <- gradientDesc(x = Methylation, k=2, max_iter=10000, plot = TRUE)
```

---

**Description**

This function estimates the vector \( \mu \), matrix A and matrix B using the optimization algorithm chosen by the user. The PDLB method allows to enter a binary matrix with missing data.

**Usage**

```r
LogBip(  
  x,  
  k = 2,  
  method = "MM",  
  type = NULL,  
  plot = TRUE,  
  maxit = NULL,  
  endsegm = 0.9,  
  label.ind = FALSE,  
  col.ind = NULL,
```

---

**Fitting a Binary Logistic Biplot using optimization methods**
draw = c("biplot", "ind", "var"),
random_start = FALSE,
L = 0,
cv_LogBip = FALSE
)

Arguments

x          Binary matrix.
k          Dimensions number. By default $k = 2$.
method      Method to be used to estimate the parameters. By default method="CG"
type        For the conjugate-gradients method. Takes value 1 for the Fletcher–Reeves update, 2 for Polak–Ribiere and 3 for Beale–Sorenson.
plot        Plot the Bootstrap Logistic Biplot.
maxit       The maximum number of iterations. Defaults to 100 for the gradient methods, and 500 without gradient.
endsegm     The segment starts at 0.5 and ends at this value. By default endsegm = 0.90.
label.ind   By default the row points are not labelled.
col.ind     Color for the rows marks.
draw        The graph to draw ("ind" for the individuals, "var" for the variables and "biplot" for the row and columns coordinates in the same graph)
random_start Logical value; whether to randomly initialize the parameters. If FALSE, algorithm will use an SVD as starting value.
L            Penalization parameter. By default $L = 0$.
cv_LogBip   Indicates if the procedure is being used for cross validation.

Details

The methods that can be used to estimate the parameters of a logistic biplot
- For methods based on the conjugate gradient use method = "CG" and
type = 1 for the Fletcher Reeves; type = 2 for Polak Ribiere; type = 3 for Hestenes Stiefel and type = 4 for Dai Yuan.
- To use the iterative coordinate descent algorithm then method = "MM".
- If the binary matrix X has missing data, use method = "PDLB". In case it’s required to estimate the row coordinates of other individuals, this method is also the most appropriate. For more details see the paper "Logistic biplot with missing data".
- To use the BFGS formula, method = "BFGS".

Value

Coordinates of the matrix A and B, threshold for classification rule. Furthermore, for the PDLB method, the imputed matrix is returned.
Methylation

Author(s)
Giovany Babativa <gbabativam@gmail.com>

References

See Also
plotBLB, pred_LB, fitted_LB

Examples

data("Methylation")
# If the binary matrix has no missing data and does not require the projection
# of supplementary individuals, you can use an coordinate descendent MM algorithm
res_MM <- LogBip(x = Methylation, method = "MM", maxit = 1000)
# If the binary matrix has missing data or requires the projection of supplementary
# individuals, use a method based on data projection with a block coordinate descent algorithm
data("Methylation")
set.seed(12345)
 n <- nrow(Methylation)
p <- ncol(Methylation)
miss <- matrix(rbinom(n*p, 1, 0.2), n, p) #I simulate some missing data
miss <- ifelse(miss == 1, NA, miss)
x <- Methylation + miss #Matrix containing missing data
out <- LogBip(x, method = "PDLB", maxit = 1000)

Methylation

Binary data set of 48 human cell lines.

Description
A dataset containing the BRCA (breast invasive carcinoma) for 48 human cell lines, where each variable is a likely cancer driver or suppressor gene. A gene is labeled as `1' when it is classified as mutated in a sample and as `0' when classified as wild type.
Usage

Methylation

Format

A data frame with 48 rows and 9 variables:

- **GSTM1**: Glutathione S-Transferase Mu 1, is a Protein Coding gene
- **C1orf70**: Chromosome 1 Open Reading Frame 70, Transmembrane Protein
- **DNM3**: Dynamin 3, is a Protein Coding gene
- **THY1**: Thy-1 Cell Surface Antigen, is a Protein Coding gene
- **ADCY4**: Adenylate Cyclase 4, is a Protein Coding gene
- **GSTT1**: Glutathione S-Transferase Theta 1, is a Protein Coding gene
- **FILIP1L**: Filamin A Interacting Protein 1 Like, is a Protein Coding gene
- **DUSP22**: Dual Specificity Phosphatase 22, is a Protein Coding gene
- **NAPRT1**: Nicotinic Acid Phosphoribosyltransferase

Source

https://www.cancerrxgene.org

---

**performanceBLB**

*Performance comparison of several estimation algorithms*

Description

This function computes the estimates of A and B matrix with several algorithms.

Usage

```
performanceBLB(xi, k = 2, L = 0, method = NULL, maxit = NULL)
```

Arguments

- **xi**: Binary matrix.
- **k**: Dimensions number. By default \( k = 2 \).
- **L**: Penalization parameter. By default \( L = 0 \).
- **method**: use value 1 for algorithms without gradient, 2 with gradient, 3 quasi-newton methods or 4 for all methods. By default method = 2.
- **maxit**: The maximum number of iterations. Defaults to 100 for the gradient methods, and 500 without gradient.
PerformanceBLB

Details
This function compares the process time and convergence of different algorithms without gradient, with gradient or quasi-newton method for estimating the parameters in a Binary Logistic Biplot

Value
data frame with method, time of process, convergence and number of evaluations

Author(s)
Giovany Babativa <gbabativam@gmail.com>

References

See Also
gradientDesc

Examples

data('Methylation')
set.seed(123456)

########### Gradient Methods
performanceBLB(xi = Methylation)
performanceBLB(xi = Methylation, maxit = 150)

########### Without Gradient Methods
performanceBLB(xi = Methylation, method = 1)
performanceBLB(xi = Methylation, method = 1, maxit = 100)

########### Quasi-Newton Methods
performanceBLB(xi = Methylation, method = 3)
performanceBLB(xi = Methylation, method = 3, maxit = 100)

########### All methods
performanceBLB(x = Methylation, method = 4)
plotBLB

Plot a Binary Logistic Biplot using a BiplotML object

Description

Plot the bootstrap binary logistic biplot and draw confidence ellipses on the individuals of an object BiplotML.

Usage

```r
plotBLB(
  x, 
  dim = c(1, 2),
  col.ind = NULL,
  col.var = "#0E185F",
  label.ind = FALSE,
  draw = c("biplot", "ind", "var"),
  titles = NULL,
  ellipses = FALSE,
  endsegm = 0.75,
  repel = FALSE,
  xylim = NULL
)
```

Arguments

- **x**: Object class BiplotML.
- **dim**: Dimensions plot. By default Dim1 and Dim2.
- **col.ind**: Color for the individuals.
- **col.var**: Color for the variables.
- **label.ind**: By default the row points are not labelled.
- **draw**: The graph to draw ("ind" for the individuals, "var" for the variables and "biplot" for the row and columns coordinates in the same graph)
- **titles**: Title for the Biplot
- **ellipses**: If ellipses=TRUE, draw confidence ellipses around the rows.
- **endsegm**: Represents where the segment of a variable ends on the logit probability scale. By default endsegm=0.75
- **repel**: Repel overlapping text labels.
- **xylim**: vector specifying the minimum and maximum of the x-axis and y-axis. For example, you can use xylim=c(-10, 10).

Details

If draw = "ind", then the biplot is plotted only for individuals and if draw = "var" then is plotted only for the variables.
Value

Returns the Biplot of the individuals and variables.

Author(s)

Giovan Babativa <gbabativam@gmail.com>

References


See Also

bootBLB

Examples

data("Methylation")
set.seed(123456)
outBLB <- bootBLB(x = Methylation, sup = TRUE, plot=FALSE)
plotBLB(x = outBLB, titles = "Methylation Logistic Biplot", ellipses = FALSE)
plotBLB(x = outBLB, titles = "Methylation LogBiplot", endsegm = 0.95)
plotBLB(x = outBLB, label.ind = TRUE, titles = "Methylation LogBiplot")

pred_LB

Predict logistic biplot and thresholds by variable

Description

Predicts the binary matrix and calculates the optimal thresholds per variable that minimize the Balanced Accuracy (BACC)

Usage

pred_LB(object, x, ncuts = 100)

Arguments

object BiplotML object
x Binary matrix.
ncuts Number of equidistant cuts between 0 and 1 that will be evaluated. By default ncuts = 100
proj_LogBip

Details

The threshold for each variable is lowered to minimize the Balanced Accuracy (BACC).

\[
BACC = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right),
\]

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives and FN is the number of false negatives.

Value

This function returns the thresholds per variable, the predicted matrix, the confusion matrix and the BACC.

Examples

data("Methylation")
LB <- LogBip(Methylation, plot = FALSE)
out <- pred_LB(LB, Methylation)

proj_LogBip

Fitting a Binary Logistic Biplot with Missing Data Using Data Projection and a Block Coordinate Descending Algorithm

Description

This function impute the missing values of a binary dataset X, and estimates the vector \( \mu \), matrix A and matrix B using data projection model with a block coordinate descending algorithm.

Usage

proj_LogBip(x, k = 2, max_iters = 1000, random_start = FALSE, epsilon = 1e-05)

Arguments

- x: binary matrix.
- k: dimensions number. By default k = 2.
- max_iters: maximum iterations.
- random_start: random initialization
- epsilon: convergence criteria

Value

Imputed X matrix and coordenates of the matrix A and B, and \( \mu \).
Author(s)
Giovan Babativa <gbabativam@gmail.com>

References

See Also
cv_LogBip

Examples

data("Methylation")
set.seed(12345)
n <- nrow(Methylation)
p <- ncol(Methylation)
miss <- matrix(rbinom(n*p, 1, 0.2), n, p) #I simulate some missing data
miss <- ifelse(miss == 1, NA, miss)
x <- Methylation + miss #Matrix containing missing data
out <- LogBip(x, method = "PDLB", maxit = 1000)

Description
This function estimates the vector \( \mu \), matrix A and matrix B using coordinate descendent MM algorithm.

Usage

sdv_MM(
  x,
  k = 2,
  iterations = 1000,
  truncated = TRUE,
  random = FALSE,
  epsilon = 1e-04
)
**Arguments**

- `x` binary matrix.
- `k` dimensions number. By default `k = 2`.
- `iterations` maximum iterations.
- `truncated` if TRUE, find the k largest singular values and vectors of a matrix.
- `random` random initialization
- `epsilon` convergence criteria

**Value**

Coordinates of the matrix A and B, and \( \mu \)

**Author(s)**

Giovany Babativa <gbabativam@gmail.com>

**References**


**See Also**

cv_LogBip

**Examples**

```r
data("Methylation")
out <- sdv_MM(x = Methylation)
```

---

**simBin**

*Multivariate binary data*

**Description**

Simulate a binary data matrix based on a latent variables model

**Usage**

`simBin(n, p, k, D, C = 1)`
**simBin**

**Arguments**

- **n**  
  number of rows
- **p**  
  number of columns
- **k**  
  number of underlying dimensions in the model
- **D**  
  sparsity control
- **C**  
  variance control

**Value**

- **X**: binary matrix
- **P**: predicted matrix
- **Theta**: matrix of natural parameters
- **A**: row markers
- **B**: column markers
- **mu**: offset term
- **D**: sparsity level
- **n**: number of rows
- **p**: number of columns

**Author(s)**

Giovany Babativa <gbabativam@gmail.com>

**See Also**

- `cv_LogBip`

**Examples**

```r
x <- simBin(n = 100, p = 50, k = 3, D = 0.5)
```
Index

* datasets
  Methylation, 9
  bootBLB, 2, 13
  cv_LogBip, 4, 15–17
  fitted_LB, 5, 5, 9
  gradientDesc, 6, 11
  LogBip, 5, 7
  Methylation, 9
  performanceBLB, 3, 7, 10
  plotBLB, 3, 7, 9, 12
  pred_LB, 5, 9, 13
  proj_LogBip, 14
  sdv_MM, 15
  simBin, 5, 16