Package: BiplotML (via r-universe)

May 8, 2026

Title Logistic Biplot Estimation Using Machine Learning Algorithms

Version 1.1.1

Date 2026-04-30

Description Implements methods for fitting logistic biplot models to multivariate binary data. The logistic biplot represents individuals as points and binary variables as directed vectors in a low-dimensional subspace; the orthogonal projection of each individual onto a variable vector approximates the expected probability that the corresponding characteristic is present. Available fitting methods include conjugate gradient algorithms, a coordinate descent Majorization-Minimization (MM) algorithm, and a block coordinate descent algorithm based on data projection that supports matrices with missing values and allows new individuals to be projected as supplementary rows without refitting the model. A cross-validation procedure is provided to select the number of latent dimensions k. References: Babativa-Marquez and Vicente-Villardon (2021) <doi:10.3390/math9162015>; Vicente-Villardon and Galindo (2006, ISBN:9780470973196).

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports optimx, RSpectra

Suggests testthat (>= 3.0.0), knitr, rmarkdown, 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] (ORCID: <`https://orcid.org/0000-0002-4989-7459`>)

Maintainer Jose Giovany Babativa-Marquez <`jgbabativam@unal.edu.co`>

Repository https://cran.r-universe.dev

Date/Publication 2026-05-08 17:47:37 UTC

RemoteUrl https://github.com/cran/BiplotML

RemoteRef HEAD

RemoteSha e7124c6c078d3600a1a1bfd358da44055b4edfe4

Contents

`cv_LogBip` *Cross-Validation for Logistic Biplot*

Description

Performs k-fold cross-validation for a logistic biplot model across a range of dimensions, enabling selection of the optimal number of latent dimensions.

Usage

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

Arguments

`data`	A binary matrix.
`k`	Integer vector of dimensions to evaluate. Default is `0:5`.
`K`	Number of folds. Default is `K = 7`.
`method`	Fitting algorithm: `"MM"` (default), `"CG"`, `"PDLB"`, or `"BFGS"`.
`type`	Update formula for the CG method (see `LogBip`).
`plot`	Logical; if `TRUE` (default), the cross-validation error curve is plotted.
`maxit`	Maximum number of iterations. Defaults to `100` for gradient methods and `2000` for the MM algorithm.

Value

A data frame with columns `k`, `cv-error` (mean cross-validation error, in percent), and `train-error` (mean training error, in percent).

Author(s)

Giovany Babativa <jgbabativam@unal.edu.co>

References

Bro, R., Kjeldahl, K., & Smilde, A. K. (2008). Cross-validation of component models: a critical look at current methods. *Analytical and Bioanalytical Chemistry*, *390*(5), 1241–1251.

Wold, S. (1978). Cross-validatory estimation of the number of components in factor and principal components models. *Technometrics*, *20*(4), 397–405.

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 using the MM algorithm
cv_MM <- cv_LogBip(data = x$X, k = 0:5, method = "MM", maxit = 1000)

# Cross-validation using the PDLB algorithm
cv_PB <- cv_LogBip(data = x$X, k = 0:5, method = "PDLB", maxit = 1000)
```

`fitted_LB` *Fitted Values for a Logistic Biplot*

Description

Computes the fitted (predicted) matrix for a logistic biplot model on either the logit (log-odds) scale or the probability scale.

Usage

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

Arguments

`object`	An object of class `BiplotML`, as returned by `LogBip`.
`type`	Scale of the fitted values: `"link"` for the logit scale (log-odds) or `"response"` for the probability scale. Partial matching is supported.

Value

A numeric matrix of fitted values with the same dimensions as the original binary matrix.

Author(s)

Giovany Babativa <jgbabativam@unal.edu.co>

Examples

```
data("Methylation")
LB    <- LogBip(Methylation, plot = FALSE)
Theta <- fitted_LB(LB, type = "link")      # log-odds scale
Pi    <- fitted_LB(LB, type = "response")  # probability scale
```

`gradientDesc` *Fit a Binary Logistic Biplot via Gradient Descent*

Description

Estimates the row-marker matrix **A** and the column-marker matrix **B** of a binary logistic biplot using a simple (batch) gradient descent algorithm. This function is mainly provided for pedagogical purposes and benchmarking; the MM and CG methods in `LogBip` are generally faster and more reliable.

Usage

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

Arguments

`x`	A binary matrix.
`k`	Number of dimensions. Default is `k = 2`.
`rate`	Learning rate α for the gradient descent update. Default is `0.001`.
`converg`	Convergence tolerance: the algorithm stops when the relative change in the loss function is below this value. Default is `0.001`.
`max_iter`	Maximum number of iterations.
`plot`	Logical; if `TRUE`, the logistic biplot is plotted after fitting. Default is `FALSE`.
`...`	Additional arguments (currently unused).

Details

The model is

$$\operatorname{logit}(\pi_{ij}) = \log\left(\frac{\pi_{ij}}{1-\pi_{ij}}\right) = \mu_j + \sum_{s=1}^{k} b_{js}\, a_{is} = \mu_j + \mathbf{a}_i^\top \mathbf{b}_j.$$

The gradient with respect to the full parameter vector is

$$\nabla\ell = \left(\frac{\partial\ell}{\partial\boldsymbol{\mu}}, \frac{\partial\ell}{\partial\mathbf{A}}, \frac{\partial\ell}{\partial\mathbf{B}}\right) = \big((\boldsymbol{\Pi}-\mathbf{X})^\top,\ (\boldsymbol{\Pi}-\mathbf{X})\mathbf{B},\ (\boldsymbol{\Pi}-\mathbf{X})^\top\mathbf{A}\big)\,.$$

Value

An object of class `BiplotML` (a named list) containing:

`Ahat` Estimated row-marker matrix.

`Bhat` Estimated column-marker matrix (including intercepts).

`method` Character string `"Gradient Descent"`.

Author(s)

Giovany Babativa <jgbabativam@unal.edu.co>

References

Vicente-Villardon, J. L., & Galindo, M. P. (2006). Logistic biplots. In M. Greenacre & J. Blasius (Eds.), *Multiple Correspondence Analysis and Related Methods* (pp. 503–521). Chapman & Hall.

See Also

`plotBLB`, `performanceBLB`

Examples

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

`LogBip` *Fit a Binary Logistic Biplot*

Description

Estimates the intercept vector μ, the row-marker matrix $\mathbf{A}$, and the column-marker matrix $\mathbf{B}$ of a logistic biplot model using the optimization algorithm selected by the user.

Usage

```
LogBip(
  x,
  k = 5,
  method = "MM",
  type = NULL,
  plot = TRUE,
  maxit = NULL,
  endsegm = 0.9,
  label.ind = FALSE,
  col.ind = NULL,
  draw = c("biplot", "ind", "var"),
  random_start = FALSE,
  L = 0,
  cv_LogBip = FALSE
)
```

Arguments

`x`	A binary matrix (or a matrix with `NA` values when `method = "PDLB"`).
`k`	Number of dimensions. Default is `k = 5`.
`method`	Fitting algorithm. One of `"MM"` (default), `"CG"`, `"PDLB"`, or `"BFGS"`.
`type`	Update formula for the conjugate gradient method: `1` = Fletcher–Reeves, `2` = Polak–Ribiere, `3` = Beale–Sorenson. Ignored for other methods.

`plot`	Logical; if `TRUE` (default), the logistic biplot is plotted after fitting.
`maxit`	Maximum number of iterations. Defaults to `100` for gradient methods and `500` for derivative-free methods.
`endsegm`	End point of the variable segment on the probability scale. The segment starts at 0.5 and ends at this value. Default is `0.90`.
`label.ind`	Logical; if `TRUE`, row points are labelled. Default is `FALSE`.
`col.ind`	Color for the row markers. Passed to `plotBLB`.
`draw`	Which graph to draw: `"biplot"` (default) for both rows and columns, `"ind"` for individuals only, or `"var"` for variables only.
`random_start`	Logical; if `TRUE`, parameters are initialised randomly. If `FALSE` (default), an SVD-based initialisation is used.
`L`	Ridge penalization parameter. Default is `L = 0` (no penalty).
`cv_LogBip`	Logical; indicates whether the function is being called internally by `cv_LogBip`. Users should leave this as `FALSE` (default).

Details

The following fitting methods are available:

Conjugate gradient (CG): Set `method = "CG"` and choose the update formula via `type`:

- `type = 1` — Fletcher–Reeves
- `type = 2` — Polak–Ribiere
- `type = 3` — Hestenes–Stiefel
- `type = 4` — Dai–Yuan

Coordinate descent MM: Set `method = "MM"` to use the iterative coordinate descent Majorization-Minimization algorithm.

Projection-based algorithm (PDLB): Set `method = "PDLB"` when the binary matrix contains missing values, or when the row coordinates of new (supplementary) individuals need to be estimated without refitting the model. See Babativa-Marquez & Vicente-Villardon (2022) for details.

BFGS: Set `method = "BFGS"` to use the Broyden–Fletcher– Goldfarb–Shanno quasi-Newton method.

Value

An object of class `BiplotML` (a named list) containing:

`Ahat` Data frame of row-marker coordinates.

`Bhat` Data frame of column-marker coordinates, including the intercept column `bb0`.

`method` Character string identifying the fitting method used.

`loss_function` Vector of loss-function values at each iteration (MM and PDLB methods only).

`iterations` Number of iterations performed (MM and PDLB methods only).

`impute_x` Imputed binary matrix (PDLB method only).

Author(s)

Giovany Babativa <jgbabativam@unal.edu.co>

References

Babativa-Marquez, J. G., & Vicente-Villardon, J. L. (2026). Logistic biplot with missing data. *In process*.

Babativa-Marquez, J. G., & Vicente-Villardon, J. L. (2021). Logistic biplot by conjugate gradient algorithms and iterated SVD. *Mathematics*, *9*(16), 2015. doi:10.3390/math9162015

Nash, J. C. (2011). Unifying optimization algorithms to aid software system users: optimx for R. *Journal of Statistical Software*, *43*(9), 1–14.

Nash, J. C. (2014). On best practice optimization methods in R. *Journal of Statistical Software*, *60*(2), 1–14.

Nocedal, J., & Wright, S. (2006). *Numerical Optimization* (2nd ed.). Springer.

Vicente-Villardon, J. L., & Galindo, M. P. (2006). Logistic biplots. In M. Greenacre & J. Blasius (Eds.), *Multiple Correspondence Analysis and Related Methods* (pp. 503–521). Chapman & Hall.

See Also

`plotBLB`, `pred_LB`, `fitted_LB`

Examples

```
data("Methylation")

# Fit using the coordinate descent MM algorithm
res_MM <- LogBip(x = Methylation, method = "MM", maxit = 1000)

# Fit using the PDLB algorithm with simulated missing data
set.seed(12345)
n <- nrow(Methylation); p <- ncol(Methylation)
miss <- matrix(rbinom(n * p, 1, 0.2), n, p)
miss <- ifelse(miss == 1, NA, miss)
x_miss <- Methylation + miss
res_PDLB <- LogBip(x = x_miss, method = "PDLB", maxit = 1000)
```

`Methylation` *DNA Methylation Binary Data*

Description

A binary matrix of DNA methylation measurements for a sample of individuals. Each row represents an individual and each column a CpG site; a value of 1 indicates methylation and 0 indicates no methylation.

Usage

```
Methylation
```

Format

A binary matrix with 50 rows (individuals) and 13 columns (CpG sites).

Source

Publicly available methylation data used for illustrative purposes.

Examples

```
data("Methylation")
dim(Methylation)
```

`performanceBLB` *Compare Optimization Algorithms for Binary Logistic Biplot Estimation*

Description

Fits the binary logistic biplot model using multiple optimization algorithms and returns a summary of their computation time, convergence status, and number of function evaluations, facilitating algorithm selection.

Usage

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

Arguments

`xi`	A binary matrix.
`k`	Number of dimensions. Default is `k = 2`.
`L`	Ridge penalization parameter. Default is `L = 0`.
`method`	Algorithm group to compare: `1` (derivative-free), `2` (gradient, default), `3` (quasi-Newton), or `4` (all).
`maxit`	Maximum number of iterations per algorithm.

Details

The following algorithm groups are available via the `method` argument:

- `1` — Derivative-free methods: Nelder-Mead, UOBYQA, NEWUOA.
- `2` — Gradient methods (default): CG, Rcgmin.
- `3` — Quasi-Newton methods: BFGS, L-BFGS-B, nlm, nlminb.
- `4` — All of the above.

Value

A data frame with one row per algorithm and columns:

`method`	Algorithm name.
`evaluat`	Final value of the objective function.
`convergence`	Convergence status.
`fevals`	Number of function evaluations.
`time`	Elapsed computation time.

Author(s)

Giovany Babativa <jgbabativam@unal.edu.co>

References

Nash, J. C. (2011). Unifying optimization algorithms to aid software system users: optimx for R. *Journal of Statistical Software*, *43*(9), 1–14.

Nash, J. C. (2014). On best practice optimization methods in R. *Journal of Statistical Software*, *60*(2), 1–14.

Vicente-Villardon, J. L., & Galindo, M. P. (2006). Logistic biplots. In M. Greenacre & J. Blasius (Eds.), *Multiple Correspondence Analysis and Related Methods* (pp. 503–521). Chapman & Hall.

See Also

`gradientDesc`

Examples

```
data("Methylation")
set.seed(123456)

# Gradient methods (default)
performanceBLB(xi = Methylation)
performanceBLB(xi = Methylation, maxit = 150)

# Derivative-free 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(xi = Methylation, method = 4)
```

`plotBLB` *Plot a Binary Logistic Biplot*

Description

Produces a **ggplot2**-based logistic biplot from a `BiplotML` object fitted with `LogBip`. Supports coloring and shaping of row markers by a categorical variable, filled arrowheads, dashed reference lines that span the full plot area, and flexible axis-limit control via `xylim`, `xlim`, and `ylim`.

Usage

```
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,
  xlim = NULL,
  ylim = NULL,
  escala = NULL
)
```

Arguments

`x`	An object of class `BiplotML`, as returned by `LogBip`.
`dim`	Integer vector of length 2 specifying which dimensions to plot. Default is `c(1, 2)`.
`col.ind`	Optional vector of the same length as the number of rows in the original data, used to color *and* shape the row markers by a categorical variable (e.g., `col.ind = df$group`). Levels are mapped to the `"Set1"` palette and to filled geometric shapes. If `NULL` (default), all row markers are drawn as gold triangles (`shape = 17`, color `"#E7B800"`) when no `col.ind` is provided.
`col.var`	Color for the variable arrows. Default is `"#0E185F"` (dark navy).
`label.ind`	Logical; if `TRUE`, row markers are labelled. Default is `FALSE`.
`draw`	Which graph to draw. One of `"biplot"` (default, both row and column markers), `"ind"` (row markers only), or `"var"` (variable arrows only). Partial matching is supported.
`titles`	Main title for the plot. If `NULL` (default), a generic title is used depending on `draw`.

`ellipses`	Logical; if `TRUE`, bootstrap confidence ellipses are drawn around the row markers. Requires a bootstrap fit
`endsegm`	End point of the variable arrow on the probability scale. The arrow starts at $p = 0.5$ and ends at this value. Default is `0.75`.
`repel`	Logical; if `TRUE`, overlapping variable labels are repelled using **ggrepel**. Default is `FALSE`.
`xylim`	Numeric vector of length 2 specifying a symmetric range applied to both axes, e.g., `c(-80, 80)`. Overrides automatic limits. Takes precedence over automatic limits but is overridden by `xlim`/`ylim` if those are also supplied. Default is `NULL`.
`xlim`	Numeric vector of length 2 specifying the range of the x-axis independently, e.g., `c(-100, 60)`. Takes precedence over `xylim`. Default is `NULL`.
`ylim`	Numeric vector of length 2 specifying the range of the y-axis independently, e.g., `c(-80, 80)`. Takes precedence over `xylim`. Default is `NULL`.
`escala`	Positive numeric scalar. Multiplicative factor applied to the row marker coordinates (`x$Ahat`) before plotting, so that they are on a comparable visual scale to the variable arrows. If `NULL` (default), the value is chosen automatically so that the range of the scaled row markers matches the range of the variable arrows, producing a visually balanced biplot. Pass an explicit numeric value to override the automatic calculation (e.g., `escala = 65`).

Details

Variable vectors are drawn as arrows from the point where the predicted probability equals 0.5 to the point where it equals `endsegm`. Short arrows indicate a rapid increase in the probability of the corresponding characteristic. The orthogonal projection of a row marker onto a variable's arrow approximates the probability that the characteristic is present for that individual.

The three arguments that control axis limits are evaluated in the following order of priority:

1. `xlim` and `ylim` (independent limits for each axis).
2. `xylim` (symmetric limits applied to both axes).
3. Automatic limits derived from all plotted elements.

The `escala` argument multiplies the row marker coordinates before plotting so that they are visually comparable to the variable arrows, which are expressed in the original parameter units. It only affects the display, not the stored coordinates.

Value

A `ggplot2` object that can be further customised with standard **ggplot2** functions (e.g., `theme()`, `labs()`).

Author(s)

Giovany Babativa <jgbabativam@unal.edu.co>

References

Meulman, J. J., & Heiser, W. J. (1983). *The Display of Bootstrap Solutions in Multidimensional Scaling* (Technical memorandum). Bell Laboratories.

Vicente-Villardon, J. L., & Galindo, M. P. (2006). Logistic biplots. In M. Greenacre & J. Blasius (Eds.), *Multiple Correspondence Analysis and Related Methods* (pp. 503–521). Chapman & Hall.

See Also

`LogBip`

Examples

```
data("Methylation")
set.seed(123456)
res <- LogBip(x = Methylation, method = "MM", maxit = 1000, plot = FALSE)
```

`pred_LB` *Predict Binary Responses from a Logistic Biplot*

Description

Predicts the binary response matrix from a fitted logistic biplot and computes the optimal classification threshold for each variable by minimising the Balanced Error Rate (BER).

Usage

```
pred_LB(object, x, ncuts = 100)
```

Arguments

`object`	An object of class `BiplotML`, as returned by `LogBip`.
`x`	The original binary matrix used to fit the model.
`ncuts`	Number of equally spaced threshold candidates in $[0, 1]$. Default is `100`.

Details

The optimal threshold for variable j is the value $\alpha_j \in [0, 1]$ that minimises the Balanced Error Rate:

$$BER_j = 1 - \frac{1}{2}\left(\frac{TP_j}{TP_j + FN_j} + \frac{TN_j}{TN_j + FP_j}\right),$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively.

Value

A named list of class `BiplotML` with components:

`thresholds`	Data frame with the optimal threshold and minimum BER for each variable.
`predictX`	Predicted binary matrix.
`fitted`	Confusion matrix (sensitivity, specificity, global accuracy) for each variable.
`BER`	Overall Balanced Error Rate (in percent).

Author(s)

Giovany Babativa <jgbabativam@unal.edu.co>

Examples

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

`proj_LogBip` *Fit a Binary Logistic Biplot with Missing Data via Block Coordinate Descent*

Description

Estimates the intercept vector μ, the row-marker matrix **A**, and the column-marker matrix **B** using a data-projection model with a block coordinate descent algorithm. Missing values in the binary matrix are imputed iteratively during model fitting. This function also allows new individuals to be projected as supplementary rows without refitting the model, since the row markers are derived directly from the estimated column markers. This is the low-level function called by `LogBip` when `method = "PDLB"`.

Usage

```
proj_LogBip(x, k = 5, max_iters = 1000, random_start = FALSE, epsilon = 1e-05)
```

Arguments

`x`	A binary matrix, possibly containing `NA` values.
`k`	Number of dimensions. Default is `k = 5`.
`max_iters`	Maximum number of iterations. Default is `1000`.
`random_start`	Logical; if `TRUE`, parameters are initialised randomly. Default is `FALSE` (SVD initialisation).
`epsilon`	Convergence tolerance for the relative decrease in the loss function. Default is `1e-5`.

Value

A named list with components:

`mu` Estimated intercept vector of length p.

`A` Estimated row-marker matrix ($n \times k$).

`B` Estimated column-marker matrix ($p \times k$).

`x_est` Imputed binary matrix (missing entries replaced by fitted values).

`iter` Number of iterations performed.

`loss_funct` Vector of normalised loss-function values at each iteration.

Author(s)

Giovany Babativa <jgbabativam@unal.edu.co>

References

Babativa-Marquez, J. G., & Vicente-Villardon, J. L. (2026). Logistic biplot with missing data. *In process*.

Babativa-Marquez, J. G., & Vicente-Villardon, J. L. (2021). Logistic biplot by conjugate gradient algorithms and iterated SVD. *Mathematics*, *9*(16), 2015. doi:10.3390/math9162015

Vicente-Villardon, J. L., & Galindo, M. P. (2006). Logistic biplots. In M. Greenacre & J. Blasius (Eds.), *Multiple Correspondence Analysis and Related Methods* (pp. 503–521). Chapman & Hall.

See Also

`LogBip`, `cv_LogBip`

Examples

```
data("Methylation")
set.seed(12345)
n <- nrow(Methylation); p <- ncol(Methylation)
miss <- matrix(rbinom(n * p, 1, 0.2), n, p)
miss <- ifelse(miss == 1, NA, miss)
x_miss <- Methylation + miss
out <- proj_LogBip(x = x_miss, k = 2, max_iters = 1000)
```

`sdv_MM` *Fit a Binary Logistic Biplot via Coordinate Descent MM Algorithm*

Description

Estimates the intercept vector μ, the row-marker matrix $\mathbf{A}$, and the column-marker matrix $\mathbf{B}$ using an iterative coordinate descent Majorization-Minimization (MM) algorithm. This is the low-level function called by `LogBip` when `method = "MM"`.

Usage

```
sdv_MM(
  x,
  k = 5,
  iterations = 1000,
  truncated = TRUE,
  random = FALSE,
  epsilon = 1e-04
)
```

Arguments

`x`	A binary matrix with no missing values.
`k`	Number of dimensions. Default is `k = 5`.
`iterations`	Maximum number of iterations. Default is `1000`.
`truncated`	Logical; if `TRUE` (default for large matrices), the truncated SVD from **RSpectra** is used to speed up computation.
`random`	Logical; if `TRUE`, parameters are initialised randomly. Default is `FALSE` (SVD initialisation).
`epsilon`	Convergence tolerance. The algorithm stops when the relative decrease in the loss function is below this value. Default is `1e-4`.

Value

A named list with components:

`mu` Estimated intercept vector of length p.

`A` Estimated row-marker matrix ($n \times k$).

`B` Estimated column-marker matrix ($p \times k$).

`iterations` Number of iterations performed.

`loss_func` Vector of normalised loss-function values at each iteration.

Author(s)

Giovany Babativa <jgbabativam@unal.edu.co>

References

Babativa-Marquez, J. G., & Vicente-Villardon, J. L. (2021). Logistic biplot by conjugate gradient algorithms and iterated SVD. *Mathematics*, *9*(16), 2015. doi:10.3390/math9162015

Vicente-Villardon, J. L., & Galindo, M. P. (2006). Logistic biplots. In M. Greenacre & J. Blasius (Eds.), *Multiple Correspondence Analysis and Related Methods* (pp. 503–521). Chapman & Hall.

See Also

`LogBip`, `cv_LogBip`

Examples

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

`simBin` *Simulate a Multivariate Binary Matrix*

Description

Simulates a binary data matrix from a logistic biplot latent variable model with known parameters, useful for benchmarking and cross-validation studies.

Usage

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

Arguments

`n`	Number of rows (individuals).
`p`	Number of columns (variables).
`k`	Number of underlying latent dimensions.
`D`	Sparsity control: the marginal probability of a 1 in the population. A value close to 0 or 1 yields a sparse or dense matrix, respectively.
`C`	Variance scaling factor for the row scores. Default is `C = 1`.

Value

A named list with components:

`X` Simulated binary matrix ($n \times p$).

`P` Matrix of true Bernoulli probabilities ($n \times p$).

`Theta` Matrix of true log-odds (natural parameters).

`A` True row-marker matrix ($n \times k$).

`B` True column-marker matrix ($p \times k$), orthonormal.

`mu` True intercept vector of length p.

`D` Observed proportion of ones in $\mathbf{X}$.

`n` Number of rows.

`p` Number of columns.

Author(s)

Giovany Babativa <jgbabativam@unal.edu.co>

See Also

`cv_LogBip`

Examples

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

Index