Package ‘lori’

December 10, 2019

Type Package

Title Imputation of Count Data using Side Information

Version 2.2.0

Maintainer Genevieve Robin <genevieve.robin@inria.fr>

Description Analysis, imputation, and multiple imputation of count data using covariates. LORI uses a log-linear model where main row and column effects are decomposed as regression terms on known covariates. A residual low-rank interaction term is also fitted. LORI returns estimates of covariate effects and interactions, as well as an imputed count table. The package also contains a multiple imputation procedure.

License GPL-3

Encoding UTF-8

LazyData true

Depends stats, data.table, rARPACK, svd

Suggests knitr, rmarkdown, testthat

RoxygenNote 6.1.1

NeedsCompilation no

Author Genevieve Robin [aut, cre]

Repository CRAN

Date/Publication 2019-12-10 22:40:05 UTC

R topics documented:

`covmat` *covmat*

Description

covmat

Usage

```
covmat(n, p, R = NULL, C = NULL, E = NULL, center = F)
```

Arguments

`n`	number of rows
`p`	number ofcolumns
`R`	nxK1 matrix of row covariates
`C`	nxK2 matrix of column covariates
`E`	(n+p)xK3 matrix of row-column covariates
`center`	boolean indicating whether the returned covariate matrix should be centered (for identifiability)

Value

the joint product of R and C column-binded with E, a (np)x(K1+K2+K3) matrix in order row1col1,row2col1,...,rowncol1, row1col2, row2col2,...,rowncolp

Examples

```
R <- matrix(rnorm(10), 5)
C <- matrix(rnorm(9), 3)
covs <- covmat(5,3,R,C)
```

`cv.lori` *selection of the regularization parameters (lambda1 and lambda2) of the lori function by cross-validation*

Description

selection of the regularization parameters (lambda1 and lambda2) of the lori function by cross-validation

Usage

```
cv.lori(Y, cov = NULL, intercept = T, reff = T, ceff = T,
  rank.max = 5, N = 5, len = 20, prob = 0.2, algo = c("alt",
  "mcgd"), thresh = 1e-05, maxit = 10, trace.it = F)
```

Arguments

Y [matrix, data.frame] abundance table (nxp)

cov [matrix, data.frame] design matris (npxq)

intercept [boolean] whether an intercept should be fitted, default value is FALSE

reff [boolean] whether row effects should be fitted, default value is TRUE

ceff [boolean] whether column effects should be fitted, default value is TRUE

rank.max [integer] maximum rank of interaction matrix, default is 2

N [integer] number of cross-validation folds

len [integer] the size of the grid

prob [numeric in (0,1)] the proportion of entries to remove for cross-validation

algo type of algorithm to use, either one of "mcgd" (mixed coordinate gradient descent, adapted to large dimensions) or "alt" (alternating minimization, adapted to small dimensions)

thresh [positive number] convergence threshold, default is 1e-5

maxit [integer] maximum number of iterations, default is 100

trace.it [boolean] whether information about convergence should be printed

Value

A list with the following elements

lambda1 regularization parameter estimated by cross-validation for nuclear norm penalty (interaction matrix)

lambda2 regularization parameter estimated by cross-validation for l1 norm penalty (main effects)

errors a table containing the prediction errors for all pairs of parameters

Examples

```
X <- matrix(rnorm(20), 10)
Y <- matrix(rpois(10, 1:10), 5)
res <- cv.lori(Y, X, N=2, len=2)
```

---

lori *main function: analysis and imputation of incomplete count data tables using side information (row-column attributes).*

Description

main function: analysis and imputation of incomplete count data tables using side information (row-column attributes).

Usage

```
lori(Y, cov = NULL, lambda1 = NULL, lambda2 = NULL, intercept = T,
  reff = T, ceff = T, rank.max = 2, algo = c("alt", "mcgd"),
  thresh = 1e-05, maxit = 100, trace.it = F)
```

Arguments

`Y` [matrix, data.frame] count table (nxp).

`cov` [matrix, data.frame] design matrix (np*q) in order row1xcol1,row2xcol2,..,rownxcol1,row1xcol2,row2xco

`lambda1` [positive number] the regularization parameter for the interaction matrix.

`lambda2` [positive number] the regularization parameter for the covariate effects.

`intercept` [boolean] whether an intercept should be fitted, default value is FALSE

`reff` [boolean] whether row effects should be fitted, default value is TRUE

`ceff` [boolean] whether column effects should be fitted, default value is TRUE

`rank.max` [integer] maximum rank of interaction matrix (smaller than min(n-1,p-1))

`algo` type of algorithm to use, either one of "mcgd" (mixed coordinate gradient descent, adapted to large dimensions) or "alt" (alternating minimization, adapted to small dimensions)

`thresh` [positive number] convergence tolerance of algorithm, by default `1e-6`.

`maxit` [integer] maximum allowed number of iterations.

`trace.it` [boolean] whether convergence information should be printed

Value

A list with the following elements

`X`	nxp matrix of log of expected counts
`alpha`	row effects
`beta`	column effects
`epsilon`	covariate effects
`theta`	nxp matrix of row-column interactions
`imputed`	nxp matrix of imputed counts
`means`	nxp matrix of expected counts (exp(X))
`cov`	npxK matrix of covariates

Examples

mi.lori *multiple imputation of count data using the lori model*

Description

multiple imputation of count data using the lori model

Usage

```
mi.lori(Y, cov = NULL, lambda1 = NULL, lambda2 = NULL, M = 25,
  intercept = T, reff = T, ceff = T, rank.max = 5,
  algo = c(”alt”, ”mcgd”), thresh = 1e-05, maxit = 1000,
  trace.it = F)
```

Arguments

`Y`	[matrix, data.frame] count table (nxp).
`cov`	[matrix, data.frame] design matrix (np*q) in order row1xcol1,row2xcol2,..,rownxcol1,row1xcol2,row2xco
`lambda1`	[positive number] the regularization parameter for the interaction matrix.
`lambda2`	[positive number] the regularization parameter for the covariate effects.
`M`	[integer] the number of multiple imputations to perform
`intercept`	[boolean] whether an intercept should be fitted, default value is FALSE
`reff`	[boolean] whether row effects should be fitted, default value is TRUE
`ceff`	[boolean] whether column effects should be fitted, default value is TRUE
`rank.max`	[integer] maximum rank of interaction matrix (smaller than min(n-1,p-1))
`algo`	type of algorithm to use, either one of "mcgd" (mixed coordinate gradient descent, adapted to large dimensions) or "alt" (alternating minimization, adapted to small dimensions)
`thresh`	[positive number] convergence tolerance of algorithm, by default `1e-6`.
`maxit`	[integer] maximum allowed number of iterations.
`trace.it`	[boolean] whether convergence information should be printed

Value

`mi.imputed`	a list of length M containing the imputed count tables
`mi.alpha`	a (Mxn) matrix containing in rows the estimated row effects (one row corresponds to one single imputation)
`mi.beta`	a (Mxp) matrix containing in rows the estimated column effects (one row corresponds to one single imputation)
`mi.epsilon`	a (Mxq) matrix containing in rows the estimated effects of covariates (one row corresponds to one single imputation)
`mi.theta`	a list of length M containing the estimated interaction matrices
`mi.mu`	a list of length M containing the estimated Poisson means
`mi.y`	list of bootstrapped count tables used fot multiple imputation
`Y`	original incomplete count table

Examples

```
X <- matrix(rnorm(50), 25)
Y <- matrix(rpois(25, 1:25), 5)
res <- mi.lori(Y, X, 10, 10, 2)
```

pool.lori *aggregate lori multiple imputation results*

Description

aggregate lori multiple imputation results

Usage

```
pool.lori(res.mi)
```

Arguments

res.mi	a multiple imputation result from the function mi.lori

Value

pool.impute	a list containing the pooled means (mean) and variance (var) of the imputed values
pool.alpha	a list containing the pooled means (mean) and variance (var) of the row effects
pool.beta	a list containing the pooled means (mean) and variance (var) of the column effects
pool.epsilon	a list containing the pooled means (mean) and variance (var) of the covariate effects
pool.theta	a list containing the pooled means (mean) and variance (var) of the interactions

Examples

```
X <- matrix(rnorm(50), 25)
Y <- matrix(rpois(25, 1:25), 5)
res <- mi.lori(Y, X, 10, 10, 2)
poolres <- pool.lori(res)
```

qut *automatic selection of nuclear norm regularization parameter*

Description

automatic selection of nuclear norm regularization parameter

Usage

```
qut(Y, cov, lambda2 = 0, q = 0.95, N = 100, reff = T, ceff = T)
```

Arguments

Y	A matrix of counts (contingency table).
cov	A (np)xK matrix of K covariates about rows and columns
lambda2	A positive number, the regularization parameter for covariates main effects
q	A number between 0 and 1. The quantile of the distribution of $lambda_QUT$ to take.
N	An integer. The number of parametric bootstrap samples to draw.
reff	[boolean] whether row effects should be fitted, default value is TRUE
ceff	[boolean] whether column effects should be fitted, default value is TRUE

Value

the value of $lambda_QUT$ to use in LoRI.

Examples

```
X = matrix(rnorm(30), 15)
Y = matrix(rpois(15, 1:15), 5)
lambda = qut(Y,X, 10, N=10)
```

Index