

Package ‘lori’

March 7, 2019

Type Package

Title Low-Rank Interactions in Count Data with Covariates

Version 2.1.2

Maintainer Genevieve Robin <genevieve.robin@polytechnique.edu>

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

Imports svd, glmnet, stats, data.table, doParallel, parallel, corpcor, foreach, FactoMineR, rARPACK

Suggests knitr, rmarkdown, testthat

RoxygenNote 6.1.0

NeedsCompilation no

Author Genevieve Robin [aut, cre]

Repository CRAN

Date/Publication 2019-03-07 06:20:09 UTC

R topics documented:

covmat	2
cv.lori	2
lori	4
mi.lori	5
plot.lori	6
pool.lori	8
qut	8

Index	10
--------------	-----------

covmat	<i>covmat</i>
--------	---------------

Description

covmat

Usage

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

Arguments

n	number of rows
p	number of columns
R	$n \times K_1$ matrix of row covariates
C	$n \times K_2$ matrix of column covariates
E	$(n+p) \times K_3$ 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 $(np) \times (K_1+K_2+K_3)$ 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	<i>selection of the regularization parameters (lambda1 and lambda2) of the lori function by cross-validation</i>
---------	--

Description

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

Usage

```
cv.lori(Y, cov = NULL, intercept = F, reff = T, ceff = T,
rank.max = 10, N = 5, len = 20, prob = 0.2, algo = c("alt",
"mcgd"), thresh = 1e-06, maxit = 100, trace.it = F, parallel = F)
```

Arguments

<code>Y</code>	[matrix, data.frame] abundance table (n x p)
<code>cov</code>	[matrix, data.frame] design matrix (n x q)
<code>intercept</code>	[boolean] whether an intercept should be fitted, default value is FALSE
<code>reff</code>	[boolean] whether row effects should be fitted, default value is TRUE
<code>ceff</code>	[boolean] whether column effects should be fitted, default value is TRUE
<code>rank.max</code>	[integer] maximum rank of interaction matrix, default is 2
<code>N</code>	[integer] number of cross-validation folds
<code>len</code>	[integer] the size of the grid
<code>prob</code>	[numeric in (0,1)] the proportion of entries to remove for cross-validation
<code>algo</code>	type of algorithm to use, either one of "mcd" (mixed coordinate gradient descent, adapted to large dimensions) or "alt" (alternating minimization, adapted to small dimensions)
<code>thresh</code>	[positive number] convergence threshold, default is 1e-5
<code>maxit</code>	[integer] maximum number of iterations, default is 100
<code>trace.it</code>	[boolean] whether information about convergence should be printed
<code>parallel</code>	[boolean] whether the N-fold cross-validation should be parallelized, default value is TRUE

Value

A list with the following elements

<code>lambda1</code>	regularization parameter estimated by cross-validation for nuclear norm penalty (interaction matrix)
<code>lambda2</code>	regularization parameter estimated by cross-validation for L1 norm penalty (main effects)
<code>errors</code>	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 = F,
     reff = T, ceff = T, rank.max = 10, algo = c("alt", "mcdg"),
     thresh = 1e-05, maxit = 1000, trace.it = F)
```

Arguments

Y	[matrix, data.frame] count table (n _{xp}).
cov	[matrix, data.frame] design matrix (n _p *q) in order row1xcol1,row2xcol2,...,rownxcol1,row1xcol2,row2xcol2,...,rownxcol2
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 "mcdg" (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	n _{xp} matrix of log of expected counts
alpha	row effects
beta	column effects
epsilon	covariate effects
theta	n _{xp} matrix of row-column interactions
imputed	n _{xp} matrix of imputed counts
means	n _{xp} matrix of expected counts (exp(X))
cov	n _p xK 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,
        prob = 0.1, reff = T, ceff = T, rank.max = 10, 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,row2xcol2,...,rownxcol2
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
prob	[positive number in (0,1)] the proportion of entries to remove in bootstrap
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 for 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)
```

plot.lori	<i>plot 2D displays of the row-column interactions</i>
-----------	--

Description

plot 2D displays of the row-column interactions

Usage

```
## S3 method for class 'lori'
plot(x, axes = c(1, 2), xlim = NULL, ylim = NULL,
     invisible = c("none", "row", "col", "row.sup", "col.sup", "quali.sup"),
     choix = c("CA", "quanti.sup"), col.row = "blue", col.col = "red",
     col.row.sup = "darkblue", col.col.sup = "darkred",
     col.quali.sup = "magenta", col.quanti.sup = "blue",
     label = c("all", "none", "row", "row.sup", "col", "col.sup",
              "quali.sup"), title = NULL, palette = NULL, autoLab = c("auto",
              "yes", "no"), new.plot = FALSE, selectRow = NULL, selectCol = NULL,
     unselect = 0.7, shadowtext = FALSE, habillage = "none",
     legend = list(bty = "y", x = "topleft"), ...)
```

Arguments

x	a lori object resulting from the lori function
axes	a vector of integers of length two, indicating which axes to plot (c(1,2) means that the first two directions are plotted).
xlim	a vector of length two indicating the limits of the plot abscissa.
ylim	a vector of length two indicating the limits of the plot ordinate.
invisible	string indicating if some points should be unlabelled ("row", "col", "row.sup", "col.sup", "quali.sup")
choix	the graph to plot ("CA" for the CA map, "quanti.sup" for the supplementary quantitative variables)
col.row	a color for the rows points
col.col	a color for columns points

col.row.sup	a color for the supplementary rows points
col.col.sup	a color for supplementary columns points
col.quali.sup	a color for the supplementary quantitative variables
col.quant.sup	a color for the supplementary quantitative variables
label	a list of character for the elements which are labelled (by default, all the elements are labelled ("row", "row.sup", "col", "col.sup", "quali.sup"))
title	string corresponding to the title of the graph you draw (by default NULL and a title is chosen)
palette	the color palette used to draw the points. By default colors are chosen. If you want to define the colors : palette=palette(c("black","red","blue")); or you can use: palette=palette(rainbow(30)), or in black and white for example: palette=palette(gray(seq(0,.9,len=25)))
autoLab	if autoLab="auto", autoLab is equal to "yes" if there are less than 50 elements and "no" otherwise; if "yes", the labels of the drawn elements are placed in a "good" way (can be time-consuming if many elements), and if "no" the elements are placed quickly but may overlap
new.plot	boolean, if TRUE, a new graphical device is created
selectRow	a selection of the rows that are drawn; see the details section
selectCol	a selection of the columns that are drawn; see the details section
unselect	may be either a value between 0 and 1 that gives the transparency of the unselected objects (if unselect=1 the transparency is total and the elements are not drawn, if unselect=0 the elements are drawn as usual but without any label) or may be a color (for example unselect="grey60")
shadowtext	boolean; if true put a shadow on the labels (rectangles are written under the labels which may lead to difficulties to modify the graph with another program)
habillage	color the individuals among a categorical variable (give the number of the categorical supplementary variable or its name)
legend	a list of arguments that defines the legend if needed (when individuals are drawn according to a variable); see the arguments of the function legend
...	further arguments passed to or from other methods, such as cex, cex.main, ...

Examples

```
X = matrix(rnorm(rep(0, 15)), 5)
Y <- matrix(rpois(length(c(X)), exp(c(X))), 5)
res_lori <- lori(Y, cov=cbind(c(X),c(X)), lambda1=1, lambda2=1)
p <- plot(res_lori)
```

`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 λ_{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 λ_{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

covmat, 2
cv.lori, 2

lori, 4

mi.lori, 5

plot.lori, 6
pool.lori, 8

qut, 8