Package: multimix (via r-universe)

August 25, 2024

Type Package

Title Fit Mixture Models Using the Expectation Maximisation (EM) Algorithm

Version 1.0-11

Date 2023-01-28

Description A set of functions which use the Expectation Maximisation (EM) algorithm (Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977) <doi:10.1111/j.2517-6161.1977.tb01600.x> Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society, 39(1), 1--22) to take a finite mixture model approach to clustering. The package is designed to cluster multivariate data that have categorical and continuous variables and that possibly contain missing values. The method is described in Hunt, L. and Jorgensen, M. (1999) <doi:10.1111/1467-842X.00071> Australian & New Zealand Journal of Statistics 41(2), 153--171 and Hunt, L. and Jorgensen, M. (2003) <doi:10.1016/S0167-9473(02)00190-1> Mixture model clustering for mixed data with missing information, Computational Statistics & Data Analysis, 41(3-4), 429--440.

Depends mvtnorm, R (>= 4.0.0)

Encoding UTF-8

Imports methods, Rcpp (>= 1.0.10)

LazyData true

License GPL (>= 2)

LinkingTo Rcpp

URL `https://github.com/jmcurran/multimix`

BugReports `https://github.com/jmcurran/multimix/issues`

RoxygenNote 7.2.3

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

RemoteUrl https://github.com/jmcurran/multimix

RemoteRef HEAD

RemoteSha bacd1417c37c772279c5cb3774904ac805170110

Contents

`cancer.df` *Prostate cancer patient data*

Description

Data on 475 prostate cancer patients

Usage

```
data(cancer.df)
```

Format

A data.frame with 475 rows and 12 columns:

age Age in years

wt Weight in pounds

pf Patient activity

hx Family history of cancer

sbp Systolic blood pressure

dbp Diastolic blood pressure

ekg Electrocardiogram code

hg Serum haemoglobin

sz Size of primary tumour

sg Index of tumour stage and histolic grade

ap Serum prostatic acid phosphatase

bm Bone metastatses

Details

There are twelve pre-trial covariates measured on each patient, seven may be taken to be continuous, four to be discrete, and one variable (SG) is an index nearly all of whose values lie between 7 and 15, and which could be considered either discrete or continuous. We will treat SG as a continuous variable.

A preliminary inspection of the data showed that the sizeof the primary tumour (SZ) and serum prostatic acid phosphatase (AP) were both skewed variables. These variables have therefore been transformed. A square root transformation was used for SZ, and a logarithmic transformation was used for AP to achieve approximate normality. (As for correlation, skewness over the whole data set does not necessarily mean skewness within clusters. But when clusters were formed, within-cluster skewness was observed for these variables.)

Observations that had missing values in any of the twelve pretreatment covariates were omitted from furtheranalysis, leaving 475 out of the original 506 observations available.

The categorical variable `Patient activity` had 4 levels: 'Normally Active', 'Bed rest below 50 or more', and 'Confined to bed'. The numbers of the 475 in these groups were 428, 32, 12, and 3. The least active two groups are grouped in our data, giving 3 groups of size 428, 32, and 15.

Source

D.P. Byar and S.B. Green 'The choice of treatment for cancer patients based on covariate information - application to prostate cancer', Bulletin du Cancer 1980: 67:477–490, reproduced in D.A. Andrews and A.M. Herzberg 'Data: a collection of problems from many fields for the student and research worker' p.261–274 Springer series in statistics, Springer-Verlag. New York.

`cmc.df` *Contraceptive Method Choice data*

Description

This dataset is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. The cases are 1473 married women who were either not pregnant or do not know if they were at the time of interview.

Usage

```
data(cmc.df)
```

Format

A data.frame with 1473 rows and 10 columns:

age	Wife's age
edu	Wife's education
eduh	Husband's education
nborn	Number of children ever born
islam	Wife's religion
working	Wife is now working?
husocc	Husband's occupation
sol	Standard-of-living index
medex	Media exposure
method	Contraceptive method used

Details

The variables 'age' (in years) and 'nborn' (ranging from 0 to 16) would normally be treated as continuous; 'nborn' is skew and might well be transformed. The remaining 8 variables are categorical.

The variables 'edu', 'eduh' and 'sol' take values '1,2,3,4', #' they are ordinal with 1 = low and 4 = high. The variable 'husocc' takes the same 4 values, but it is not clear if the order has any significance.

The variables 'islam', 'working', and 'medex' are binary-valued with 0=Non-Islam, 1=Islam for 'islam'; 0=Yes, 1=No for 'working'; and 0=Good, 1=Not good for 'medex'.

The variable 'method' is ternary: 1=No-use, 2=Long-term, 3=Short-term.

Source

Tjen-Sien Lim 'Contraceptive Method Choice' 1997, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

`count.unique` *Count the number of unique items ion a vector x*

Description

Count the number of unique items ion a vector x

Usage

```
count.unique(x)
```

Arguments

`x`	a vector

Value

the number of unique items in `x`.

Author(s)

Murray Jorgensen

Examples

```
x = c(1, 2, 3)
count.unique(x)

x = c(1, 1, 1, 2, 3)
count.unique(x)
```

`data_organise` *Prepare data for use with multimix*

Description

Prepare data for use with multimix

Usage

```
data_organise(
  dframe,
  numClusters,
  numIter = 1000,
  cdep = NULL,
  lcdep = NULL,
  minpstar = 1e-09
)
```

Arguments

`dframe`	a data frame containing the data set you wish to model.
`numClusters`	the clusters you wish to fit.
`numIter`	the maximum number of steps to that the EM agorithm will run before terminating.
`cdep`	a list of multivariate normal cells.
`lcdep`	a list of location cells.
`minpstar`	Minimum denominator for application of Bayes Rule.

Value

An object of class `multimixSettings` which is a `list` with the following elements:

- `cdep` — a list of multivariate normal cells.
- `clink` — column numbers of univariate normal variables.
- `cprods` — a list over MVN cells containing a matrix of pair-wise products of columns in the cell, columns ordered by `pair.index`.
- `cvals` — a list over MVN cells containing a matrix of columns of variables in the cell
- `cvals2` — a list over MVN cells containing a matrix of squared columns of variables in the cell
- `dframe` — the `data.frame` of variables
- `discvar` — logical: the variable is takes values of either `TRUE` or `FALSE`
- `dlevs` — for discrete cells: number of levels
- `dlink` — column numbers of univariate discrete variables
- `dvals` — a list over discrete cells of level indicator matrices
- `lc` — logical: is continuous variable belonging to OT cell `TRUE/FALSE`
- `lcdep` — a list of OT cells
- `lcdisc` — column numbers of discrete variables in OT cells
- `lclink` — column numbers of continuous variables in OT cells
- `lcprods` — a list over OT cells containing a matrix of pair-wise products of continuous columns in the cell, columns ordered by `pair.index`
- `lcvals` — a list over OT cells containing a matrix of continuous columns of variables in the cell
- `lcvals2` — a list over OT cells containing a matrix of squared continuous columns of variables in the cell
- `ld` — logical: is discrete variable belonging to OT cell `TRUE/FALSE`
- `ldlevs` — for discrete variables in OT cells: number of levels
- `ldlink` — a column numbers of OT discrete variables
- `ldvals` — a list over OT cells of level indicator matrices
- `ldxc` — a list over OT cells whose members are lists over levels of matrices of the cell continuous variables whose columns are multiplied by the level indicator column
- `mc` — logical: is continuous variable not in OT cell `TRUE/FALSE`
- `md` — logical: is discrete variable not in OT cell `TRUE/FALSE`
- `minpstar` — minimum denominator for appliction of Bayes' Rule
- `n` — number of observations
- `numIter` — the maximum number of steps to that the EM agorithm will run before terminating
- `oc` — logical: is continuous variable in univariate cell `TRUE/FALSE`
- `olink` — column numbers of continuous univariate cells
- `op` — `length(olink)`
- `ovals` — `n` by `op` matrix of continuous univariate variables
- `ovals2` — `n` by `op` matrix of squared continuous univariate variables
- `numClusters` — the number of clusters in the model.

Author(s)

Murray Jorgensen

Examples

```
data(cancer.df)
D = data_organise(cancer.df, numClusters = 2)
```

`eStep` *The E(xpectation) step*

Description

The E(xpectation) step

Usage

```
eStep(P, D)
```

Arguments

`P`	an object of class `multimixParamList`–see `initParamList` for more information.
`D`	an object of class `multimixSettings`—see `data_organise` for more information.

Value

a `list` containing two elements: a `matrix` named `Z`—see `mStep` for more information, and a scalar `llik` containing the current value of the log-likelihood.

Author(s)

Murray Jorgensen

`hello` *Hello, World!*

Description

Prints 'Hello, world!'.

Usage

```
hello()
```

Examples

```
hello()
```

`initParamList` *Initialise the parameter list.*

Description

Although the starting parameter list `P` may be specified directly, Note also that any matrices specified must be positive definite. This function calculates an initial `P` from `D` and a starting value for `Z`.

Usage

```
initParamList(D, Z)
```

Arguments

`D`	an object of class `multimixSettings`—see `data_organise` for details.
`Z`	an $n \times q$ matrix, where n is the number of rows of `dframe` and q is the number of components in the mixture. During the fitting Z_{ij} holds the currently estimated probability that observation i belongs to component j. Often `Z` is initialized to a matrix of indicator columns for a partition of the data. It is also common to initialize `Z` to be the final `Z` from the fitting of a simpler model.

Value

an object of class `multimixParamList` which is a `list` with the following elements:

- `dstat` – `list` of matrices for each discrete variable not included in a location model. The matrix for each discrete variable is made up of a column of length q for each level (value) of the variable giving the expected proportion of each level (column) for each mixture component (row). Rows sum to 1.
- `ldstat` – `list` of matrices for each discrete variable within a location model. The matrix for each discrete variable is made up of a column of length q for each level (value) of the variable giving the expected proportion of each level (column) for each mixture component (row). Rows sum to 1.
- `ostat` – `matrix` with a column for each continuous variable outside any location mode whose q rows give the current estimated mean for each mixture component.
- `ostat2` – `matrix` with a column for each continuous variable outside any location mode whose $q$rows give the current estimated mean square for each mixture component.
- `osvar` – `matrix` with a column for each continuous variable outside any location mode whose q rows give the current estimated variance for each mixture component.
- `cstat` – `list` with a member for each nontrivial, fully continuous, partition cell, that is not including discrete cells or cells listed in lcdep, each member being a `matrix` with a column for each continuous variable in that cell, whose q rows give the current estimated mean for each mixture component.
- `cstat2` – `list` with a member for each nontrivial, fully continuous, partition cell, each member being a `matrix` with a column for each continuous variable in that cell, whose q rows give the current estimated mean square for each mixture component.

- `cvar` – `list` with a member for each nontrivial, fully continuous, partition cell, each member being a `matrix` with a column for each continuous variable in that cell, whose q rows give the current estimated variance for each mixture component.
- `cpstat` – `list` with a member for each nontrivial, fully continuous, partition cell, each member being the `matrix` with rows for each of the q mixture components and columns for each pair of continuous variables in that cell, as ordered by `pair.index`. The matrix elements are the currently expected products of the variable pairs arranged by component and pair.
- `ccov` – `list` with a member for each nontrivial, fully continuous, partition cell, each member being the `matrix` with rows for each of the q mixture components and columns for each pair of continuous variables in that cell, as ordered by pair.index. The matrix elements are the currently expected covariances of the variable pairs arranged by component and pair.
- `MVMV` – `list` with a member for each nontrivial, fully continuous, partition cell, each member being a list with members for each of the q mixture components whose values are the covariance matrix estimates for that cell and component.
- `lcstat` – `list` with a member for location partition cell, each member being a `matrix` with a column for each continuous variable in that cell, whose q rows give the current estimated mean for each mixture component.
- `lcstat2` – `list` with a member for location partition cell, each member being a `matrix` with a column for each continuous variable in that cell, whose q rows give the current estimated mean square for each mixture component.
- `lcpstat` – `list` with a member for each location cell, each member being the `matrix` with rows for each of the q mixture components and columns for each pair of continuous variables in that cell, as ordered by `pair.index`. The matrix elements are the currently expected products of the variable pairs arranged by component and pair.
- `lccov` – `list` with a member for each location cell, each member being the matrix with rows for each of the q mixture components and columns for each pair of continuous variables in that cell, as ordered by `pair.index`. The matrix elements are the currently estimated covariances of the variable pairs arranged by component and pair.
- `ldxcstat` – `list` with a member for each location partition cell, each member being a list with a member for each level of the cell's discrete variable that member being a matrix of mean values of the continuous variables for each level-class combination.
- `pistat` – `vector` containing estimates of population proportion in each cluster; column means of Z matrix.
- `W` – `matrix` of weights of observation i in cluster j; the columns of the Z matrix are each multiplied by constant to give W columns summing to 1.

`left` *Map integer index N>0 back to left member of generating pair.*

Description

Map integer index N>0 back to left member of generating pair.

Usage

```
left(N)
```

Arguments

`N`	positive integer scalar

Value

positive integer scalar

Author(s)

Murray Jorgensen

Examples

```
left(131)
left(57)
```

`make_Z_discrete` *Make initial Z matrix from initial assignment of observations to clusters*

Description

Z is an n by $numClusters$ matrix of non-negative numbers whose rows sum to 1. The ij^{th} element z_{ij} is a probability that observation i belongs to cluster j. Rather than begin from an initial assignment Multimix allows for a weighted assignment accross several clusters.

Usage

```
make_Z_discrete(d)
```

Arguments

`d`	integer

Details

This function yields a 0/1 valued matrix.

Value

a `matrix` whose entries are non-negative, and whose entries sum to 1.

Author(s)

Murray Jorgensen

Examples

```
stage = scan(file = system.file('extdata', 'Stage.txt', package = 'multimix'))
stage = stage - 2
Z = make_Z_discrete(stage)
```

`make_Z_fortran` *Read Z from FORTRAN output. Make into R matrix*

Description

The FORTRAN version of Multimix produces two output files: GENERAL.OUT and GROUPS.OUT. The latter mainly contains the Z matrix.

Usage

```
make_Z_fortran(gr.out = "groups.out")
```

Arguments

`gr.out`	string containing a file name.

Details

This function facilitates the obtaining of Multimix R output given Multimix FORTRAN output.

Value

a matrix containing a Z matrix.

Author(s)

Murray Jorgensen

Examples

```
Z <- make_Z_fortran(system.file('extdata', 'GROUPS-BP-Multimixf90.OUT',
                    package = 'multimix'))
```

`make_Z_random` *Start from random groups of similar size.*

Description

A large number (n) of observations are assigned randomly into (xq) clusters. It is recommended to repeat Multimix runs with a number of different seeds to search for a log-likelihood maximum.

Usage

```
make_Z_random(D, seed = NULL)
```

Arguments

`D`	an object of class `multimixSettings` – see `data_organise` for more information.
`seed`	a positive integer to use as a random number seed.

Details

Also consider making additional clusters from observations with low probabilities of belonging to any cluster in a previous clustering.

Value

a matrix of dimension $n \times q$ where n is the number of observations in `D$dframe` and q is the number of clusters in the model as specified by `D$numClusters`.

Examples

```
data(cancer.df)
D = data_organise(cancer.df, numClusters = 2)
Z = make_Z_random(D)
table(Z)
```

`mmain` *Title*

Description

Title

Usage

```
mmain(D, Z, P, eps = 1e-09)
```

Arguments

`D`	an object of class `multimixSettings` - see `data_organise` for full description.
`Z`	a matrix
`P`	a matrix
`eps`	Minimum increase in loglikelihood per EM step. If this is not exceeded the the algorithm will terminate.

Value

an object of class `multimix results` which is a a list containing four elements: the `multmixSettings` object `D`, the Z matrix, the P matrix, and a results matrix, called `results`, with n rows and $numClusters$ columns.

Author(s)

Murray Jorgensen

Examples

```
data(cancer.df)
D <- data_organise(cancer.df, numClusters = 2)
stage <- scan(system.file('extdata', 'Stage.txt', package = 'multimix')) - 2
Z <- make_Z_discrete(stage)
P <- initParamList(D,Z)
zpr <- mmain(D,Z,P)
zpr
```

`mStep` *The M(aximisation) step*

Description

Uses the current group membership to estimate the probabilities.

Usage

```
mStep(Z, D)
```

Arguments

`Z`	an $n \times q$ matrix, where n is the number of rows of `dframe` and q is the number of components in the mixture. During the fitting Z_{ij} holds the currently estimated probability that observation i belongs to component j. Commonly `Z` is initialized to a matrix of indicator columns for a partition of the data.
`D`	an object of class `multimixSettings`—see `data_organise` for details.

Value

an object of class `multimixParamList`—see `initParamList` for more information.

Author(s)

Murray Jorgensen

`multimix` *multimix Model-based clustering using the EM (Expectation Maximisation) algorithm*

Description

The package provides three categories of important functions:

- operational – these functions are the functions used to perform model fitting.
- helper – these functions are called internally and are unlikely to be called directly. They may not be exported in future versions of multimix.
- S3 methods – this set of functions helps with the display (printing or plotting) of the either the inputs or the results.

multimix operational functions

`data_organise` `make_Z_discrete` `make_Z_fortran` `make_Z_random` `initParamList` `mmain` `eStep` `mStep`

multimix helper functions

`count.unique` `left` `pair.index` `right`

multmix S3 methods

`plot.multimixResults` `print.multimixParamList` `print.multimixResults`

`pair.index` *Maps integer pairs (u,v) with 0<u<v bijectively to positive integers.*

Description

Used to reduce array dimensions by replacing A(x,y,z) by A*(x,pair.index(y,z))

Usage

```
pair.index(u, v)
```

Arguments

`u`	positive integer scalar
`v`	positive integer scalar

Value

integer scalar

Author(s)

Murray Jorgensen

Examples

```
pair.index(11,17)
pair.index(2,12)
```

`plot.multimixResults` *S3 method for plotting multimix results objects*

Description

S3 method for plotting multimix results objects

Usage

```
## S3 method for class 'multimixResults'
plot(x, ...)
```

Arguments

`x`	an object of class `multimixResults` – see `mmain` for more information.
`...`	any other arguments to be passed to `plot`. Note that because there are two calls to plot, the ... arguments will be passed to each call, and it is unlikely that this will have the desired effect.

Value

No return value, called for side effects.

Author(s)

James Curran

`print.multimixParamList`

S3 printing method for for multimix parameter results

Description

S3 printing method for for multimix parameter results

Usage

```
## S3 method for class 'multimixParamList'
print(
  x,
  type = c("means", "vars"),
  byLevel = FALSE,
  digits = c(4, 2, 3, 16),
  pedantic = FALSE,
  raw = FALSE,
  ...
)
```

Arguments

`x`	an object of class `multimixParamResults` – see `initParamList` for more information.
`type`	the statistic you want displayed. If `means` then the cluster means will be displayed for each univariate continuous variable, the cluster proportions for each level of a categorical variable, and the mean vector for each cluster and each multivariate normal variable.
`byLevel`	if `TRUE` then location model summary stats will be printed by the level of the factor in the location model. Otheriwse (default), they will be printed cluster by cluster.
`digits`	a vector of length 4. The first value determines how many decimal places to round categorical proportions to. The second value determines how many significant digits to display means to, and the third how many siginificant digits to display variances to. By default proportions are rounded to 4 decimal places, means 2 significant digits, and variances 3 significant digits. The fourth value is only used if `pedantic` == `TRUE`, and is set to 16 significant figures by default.

`pedantic`	if `TRUE` then the results are printed to high precision for checking purposes. This means `digits[4]` which is 16 decimal places by default.
`raw`	if `TRUE` then switches off all of the customised printing and uses the default print methods for `lists` etc.
`...`	additional arguments passed to `print`.

Value

No return value, called for side effects.

Author(s)

James Curran

`print.multimixResults` *S3 method for the printing of multimix results*

Description

S3 method for the printing of multimix results

Usage

```
## S3 method for class 'multimixResults'
print(x, n = FALSE, ...)
```

Arguments

`x`	an object of class `multimixResults`—see `mmain` for a description.
`n`	display the last few iterations of the cluster probabilities. If `TRUE` then the last 5 iterations will be displayed by default. Alternatively, a positive integer can be supplied. If this exceeds the number of actual iterations, the output will be truncated.
`...`	other parameters passed to `print`. Not currently used.

Value

No return value, called for side effects.

Author(s)

James Curran

`rcpp_hello` *Hello, Rcpp!*

Description

Returns an R `list` containing the character vector `c("foo", "bar")` and the numeric vector `c(0, 1)`.

Usage

```
rcpp_hello()
```

Examples

```
rcpp_hello()
```

`right` *Map integer index N>0 back to right member of generating pair.*

Description

Map integer index N>0 back to right member of generating pair.

Usage

```
right(N)
```

Arguments

`N`	positive integer scalar

Value

positive integer scalar

Author(s)

Murray Jorgensen

Examples

```
right(131)
right(57)
```

Index