Package ‘mixedMem’

December 1, 2020

Type Package

Title Tools for Discrete Multivariate Mixed Membership Models

Version 1.1.2

Date 2020-11-28

Maintainer Y. Samuel Wang <ysamuelwang@gmail.com>

Description Fits mixed membership models with discrete multivariate data (with or without repeated measures) following the general framework of Erosheva et al (2004). This package uses a Variational EM approach by approximating the posterior distribution of latent memberships and selecting hyperparameters through a pseudo-MLE procedure. Currently supported data types are Bernoulli, multinomial and rank (Plackett-Luce). The extended GoM model with fixed stayers from Erosheva et al (2007) is now also supported. See Airoldi et al (2014) for other examples of mixed membership models.

License GPL (>= 2)

Depends R (>= 3.0.2)

Imports Rcpp (>= 0.11.3), gtools

LazyData TRUE

LinkingTo Rcpp (>= 0.11.3), RcppArmadillo, BH

VignetteBuilder knitr

Suggests knitr, xtable,

NeedsCompilation yes

Author Y. Samuel Wang [aut, cre],
Elena A. Erosheva [aut]

Repository CRAN

Date/Publication 2020-12-01 17:30:13 UTC

R topics documented:

mixedMem-package .................................................. 2
ANES ................................................................. 4
The mixedMem package contains tools for fitting and interpreting discrete multivariate mixed membership models following the general framework outlined in Erosheva et al 2004. In a mixed membership models, individuals can belong to multiple groups instead of only a single group (Airoldi et al 2014). This extension allows for a richer description of heterogeneous populations and has been applied in a wide variety of contexts including: text data (Blei et al 2003), genotype sequences (Pritchard et al 2000), ranked data (Gormley and Murphy 2009), and survey data (Erosheva et al 2007, Gross and Manrique-Vallier 2014).

Mixed membership model objects can be created using the `mixedMemModel` constructor function. This function checks the internal consistency of the data/parameters and returns an object suitable for use by the `mmVarFit` function. The `mmVarFit` function is the main function in the package. It utilizes a variational EM algorithm to fit an approximate posterior distribution for the latent variables and select pseudo-MLE estimates for the global parameters. A step-by-step guide to using the package is detailed in the package vignette "Fitting Mixed Membership Models using mixedMem".

The package supports multivariate models (with or without repeated measurements) where each variable can be of a different type. Currently supported data types include: Bernoulli, rank (Plackett-Luce) and multinomial. Given a fixed number of sub-populations K, we assume the following generative model for each mixed membership model:

- For each individual i = 1, ..., Total:
  - Draw $\lambda_i$ from a Dirichlet($\alpha$). $\lambda_i$ is a vector of length K whose components indicates the degree of membership for individual i in each of the K sub-populations.
  - For each variable j = 1, ..., J:
  - For each of replicate r = 1, ..., $R_j$:
For each ranking level \( n = 1, \ldots, N_{i,j,r} \):

* Draw \( Z_{i,j,r,n} \) from a multinomial(1, \( \lambda_i \)). The latent sub-population indicator \( Z_{i,j,r,n} \) determines the sub-population which governs the response for observation \( X_{i,j,r,n} \).

This is sometimes referred to as the context vector because it determines the context from which the individual responds.

* Draw \( X_{i,j,r,n} \) from the latent sub-population distribution parameterized by \( \theta_{j,Z_{i,j,r,n}} \). The parameter \( \theta \) governs the observations for each sub-population. For example, if variable \( j \) is a multinomial or rank distribution with \( V_j \) categories/candidates, then \( \theta_{j,k} \) is a vector of length \( V_j \) which parameterizes the responses to variable \( j \) for sub-population \( k \). Likewise, if variable \( j \) is a Bernoulli random variable, then \( \theta_{j,k} \) is a value which determines the probability of success.

Author(s)
Y. Samuel Wang <ysamuelwang@gmail.com>, Elena Erosheva <erosheva@uw.edu>

References


National Election Studies, 1983 Pilot Election Study. Ann Arbor, MI: University of Michigan, Center for Political Studies, 1999


Examples

```r
library(mixedMem)
data(ANES)
# Dimensions of the data set: 279 individuals with 19 responses each
dim(ANES)
# The 19 variables and their categories
# The specific statements for each variable can be found using help(ANES)
# Variables titled EQ are about Equality
```
# Variables titled IND are about Economic Individualism
# Variables titled ENT are about Free Enterprise
colnames(ANES)

# Distribution of responses
table(unlist(ANES))

# Sample Size
Total <- 279
# Number of variables
J <- 19
# we only have one replicate for each of the variables
Rj <- rep(1, J)
# Nijr indicates the number of ranking levels for each variable.
# Since all our data is multinomial it should be an array of all 1s
Nijr <- array(1, dim = c(Total, J, max(Rj)))
# Number of sub-populations
K <- 3
# There are 3 choices for each of the variables ranging from 0 to 2.
Vj <- rep(3, J)
# we initialize alpha to .2
alpha <- rep(.2, K)
# All variables are multinomial
dist <- rep("multinomial", J)
# obs are the observed responses. it is a 4-d array indexed by i,j,r,n
# note that obs ranges from 0 to 2 for each response
obs[ , ,1,1] <- as.matrix(ANES)

# Initialize theta randomly with Dirichlet distributions
set.seed(123)
theta <- array(0, dim = c(J,K,max(Vj)))
for(j in 1:J)
{
    theta[j, , ] <- gtools::rdirichlet(K, rep(.8, Vj[j]))
}

# Create the mixedMemModel
# This object encodes the initialization points for the variational EM algorithm
# and also encodes the observed parameters and responses
initial <- mixedMemModel(Total = Total, J = J, Rj = Rj,
                         Nijr = Nijr, K = K, Vj = Vj, alpha = alpha,
                         theta = theta, dist = dist, obs = obs)

## Not run:
# Fit the model
out <- mmVarFit(initial)
summary(out)

## End(Not run)
Description

ANES contains political ideology survey data from the 1983 American National Election Survey Pilot. Respondents were read a value statement and asked to report their level of agreement: ‘strongly agree’, ‘agree’, ‘can’t decide’, ‘disagree’, and strongly disagree’. The 279 complete responses to the 19 statements selected by Feldman (1988) and reanalyzed by Gross and Manrique-Vallier (2014) are included. ‘Strongly disagree’ and ‘disagree’ as well as ‘strongly agree’ and ‘agree’ have been collapsed into single categories. 0 indicates ‘agree’, 1 indicates ‘can’t decide’, and 2 indicates ‘disagree’. The statements have been grouped into 3 overarching themes as indicated in the variable names- Equality (EQ), Economic Individualism (IND), and Free Enterprise (ENT). Variable ID’s from the original ANES questionnaire are included in parentheses. An example analysis of the data is included in the "Fitting Mixed Membership Models Using mixedem" vignette.

Usage

data(ANES)

Format

A data frame with 279 individuals and 19 variables:

EQ1  If people were treated more equally in this country, we would have many fewer problems (V832169)
EQ2  We should give up on the goal of equality, since people are so different to begin with (V832172)
EQ3  Our society should do whatever is necessary to make sure that everyone has an equal opportunity to succeed (V832175)
EQ4  Some people are just better cut out than others for important positions in society (V832178)
EQ5  Some people are better at running things and should be allowed to do so (V832250)
EQ6  All kinds of people should have an equal say in running this country, not just those who are successful (V832253)
EQ7  One of the big problems in this country is that we don’t give everyone an equal chance (V832256)
IND1  Any person who is willing to work hard has a good chance of succeeding (V832170)
IND2  Hard work offers little guarantee of success (V832173)
IND3  Most people who don’t get ahead should not blame the system; they really have only themselves to blame (V832176)
IND4  Even if people are ambitious, they often cannot succeed (V832251)
IND5  If people work hard, they almost always get what they want (V832254)
IND6  Even if people try hard, they often cannot reach their goals (V832257)
ENT1  The less government gets involved with business and the economy, the better off this country will be (V832171)
ENT2  There are many goods and services that would never be available to ordinary people without governmental intervention (V832174)
ENT3  There should be no government interference with business and trade (V832177)
ENT4  Putting government regulations on business does not endanger personal freedom (V832252)
Government intervention leads to too much red tape and too many problems (V832255)

Contrary to what some people think, a free enterprise system is not necessary for our form of government to survive (V832258)

References


National Election Studies, 1983 Pilot Election Study. Ann Arbor, MI: University of Michigan, Center for Political Studies, 1999

---

**computeBIC**

*Compute the approximate BIC*

**Description**

`computeBIC` computes the approximate BIC of a given `mixedMemModel`, where the lower bound on the log-likelihood (also called ELBO) is used instead of the intractable true log-likelihood.

**Usage**

```r
computeBIC(model)
```

**Arguments**

- `model` the `mixedMemModel` object for which the BIC will be calculated.

**Details**

\[
BIC = -2ELBO + p \log(Total)
\]

where \( p \) is the number of estimated parameters and Total is the number of individuals in the sample.

This BIC model selection criteria is used in Erosheva et al (2007). The number of estimated parameters \( P \) includes the parameters \( \theta \) and \( \alpha \), but omits the variational parameters \( \phi \) and \( \delta \).

**Value**

`computeBIC` returns the approximate BIC value, a real number.

**References**

computeELBO

Compute a lower bound on the log-likelihood (ELBO)

Description

computeELBO computes the variational lower bound on the log-likelihood, also called the ELBO, for a mixed membership model.

Usage

computeELBO(model)

Arguments

model a mixedMemModel object created by the mixedMemModel constructor.

Details

The lower bound (ELBO) is the objective function in the variational EM algorithm. It is a function of the latent variables ($\phi$ and $\delta$) and the parameters ($\alpha$ and $\theta$) that be derived from Jensen’s inequality:

$$E_Q \log [p(X,Z,\Lambda)] - E_Q \log [Q(Z,\Lambda|\phi,\delta)] \leq \log P(\text{obs}|\alpha,\theta)$$

Value

computeELBO returns the lower bound on the log-likelihood, a real number.

findLabels

Mixed Membership Post-Processing

Description

findLabels finds the optimal permutation of labels that minimizes the weighted squared difference between the arrays of subpopulation parameters from a fitted mixed membership model, $\theta$ and a given comparison model.

Usage

findLabels(model, comparison, exhaustive = TRUE)

Arguments

model the fitted mixedMemModel object.

comparison an array of the same dimensions as model$theta which contains the subpopulation parameters from another model. findLabels will return a permutation of the labels of model which match to comparison most closely.

exhaustive a boolean for whether an exhaustive search should be performed. If false, a greedy algorithm is used instead.
Details

Mixed Membership models are invariant to permutations of the sub-population labels; swapping the names of each sub-population yields an equivalent model. The ordering of the labels in a fitted model is dependent on the initialization points of the variational EM algorithm. The function `findLabels` selects a permutation of the sub-population labels that best matches a given comparison model by minimizing the weighted squared difference between the $\theta$ arrays. The weights are determined by the relative frequencies of each group.

$$\text{Loss} = \sum_j \sum_k \alpha_k / \alpha_0 \left[ \sum_v (\hat{\theta}_{k,v} - \theta_{k,v})^2 \right]$$

where $\alpha_0 = \sum_k \alpha_k$

If $K$, number of sub-populations, is small, the method searches through all $K!$ permutations of the sub-population labels and select the permutation which minimizes the loss. If $K$ is large, a greedy algorithm can be used instead. This algorithm selects the best match for each fitted sub-population starting with the group with the largest fitted relative frequency.

Value

`findLabels` returns a list with two objects: `perm` and `loss`. `perm` is the optimal permutation of the labels with respect to the squared error loss. `loss` is the calculated value of the weighted squared error loss (shown above) for the optimal permutation.

See Also

`permuteLabels`

Examples

```r
## Not run:
# See mixedMemModel documentation for how to generate data and instantiate a mixedMemModel object
# After the data as been generated, we initialize the array of sub-population parameters (theta)
# according to a permutation of the true labeling
set.seed(123)
perm = sample.int(K, size = K, replace = FALSE)
theta1 = theta_truth[,perm,]
test_model <- mixedMemModel(Total = Total, J = J,Rj = Rj, Nijr= Nijr, K = K, Vj = Vj,dist = dist,
obs = obs, alpha = alpha, theta = theta1)
out <- mmVarFit(test_model)
opt.perm <- findLabels(out, theta_truth)
opt.perm

# produce mixedMemModel object with sub-population labels permuted to best match
# the comparison model
out = permuteLabels(out, opt.perm$perm)

## End(Not run)
```
Description

gmv_theta contains the point estimates for the sub-population response probabilities, $\theta$ from the original Gross and Manrique-Vallier 2014 analysis. It is a 3 by 19 by 3 dimensional array with the first dimension corresponding to each latent sub-population, the second dimension corresponding to each variable, and the 3rd dimension corresponding to response choices (Agree, Can’t Decide, Disagree). Gross and Manrique-Vallier do not report results for group 3 for reasons discussed in the vignette so all values for group 3 in theta[1,] are 0.

Usage

data(gmv_theta)

References


mixedMemModel

Constructor for a Mixed Membership Model Object

Description

Constructor for a mixedMemModel object which can be used for analysis in the mixedMem package.

Usage

mixedMemModel(Total, J, Rj, Nijr, K, Vj, alpha, theta, phi = NULL, delta = NULL, dist, obs, fixedObs = NULL, P = NULL, beta = NULL)

Arguments

Total the number of individuals in the sample.
J the number of variables observed on each individual.
Rj a vector of length J specifying the number of repeated measurements for each variable.
Nijr an array with dimension (Total, J, max(Rj)) indicating the number of ranking levels for each variable and each replication. For multinomial and Bernoulli variables, Nijr[i,j,r] = 1. For rank variables, Nijr[i,j,r] indicates the number of alternatives ranked.
mixedMemModel

K
the number of sub-populations.

Vj
a vector of length J specifying the number of possible candidates for each variable. For a Bernoulli variable Vj[j] = 1.

alpha
a vector of length K which is the parameter for Dirichlet membership distribution.

theta
an array with dimensions (J,K,max(Vj)) which governs the variable distributions. The parameter theta[j,k,\] governs the distribution of variable J for a complete member of sub-population k. For instance, if variable j is a Bernoulli variable, theta[j,k,1] is the probability of success; if variable j is a multinomial variable, theta[j,k,1:Vj[j]] is the probability for each of the Vj[j] categories; if variable j is a rank variable, theta[j,k,1:Vj[j]] are the support parameters for each of the Vj[j] alternatives. Since the dimension of the relevant parameters can differ across variables, any unused elements of the array should be set to 0, while all other elements should be positive.

phi
an array with dimensions (Total,K) which specifies the variational parameters for the membership vectors, \lambda. The default group membership initialization is uniform across all groups (phi[i,k] = 1/K for all k). The default initialization is highly recommended.

delta
an array with dimensions (Total,J,max(Rj), max(Nijr),K) which specifies the variational parameters for the context vectors Z. The default initialization is uniform across all sub-populations (delta[i, j, r, n, k] = 1/K for all k).

dist
a vector of length J specifying variable types. Options are "bernoulli", "multinomial" or "rank" corresponding to the distributions of the observed variables.

obs
an array with dimensions (Total,J,max(Rj), max(Nijr)) corresponding to the observed data. For Bernoulli random variables, the data consist of 0/1’s. For multinomial or rank data the data consist of integers 0,1,...,(Vj[j]-1).

fixedObs
an array with dimensions (1,J,max(Rj), max(Nijr)) corresponding to the observed responses for a fixed group in the extended GoM model from Erosheva et al (2007)

P
scalar between 0 and 1 corresponding to initial value for the proportion of individuals in the fixed group

beta
scalar between 0 and 1 corresponding to the initial value of beta, the conditional probability of being in the fixed group for an individual who’s observed responses match the fixed group.

Details
The function returns an object of mixedMemModel class. This object contains dimensions of the model, the observed data, and the model parameters. Once a mixedMemModel object is created, the specified model can be fit for the data using the mmVarFit function. If the inputs are inconsistent (ie if dimensions do not match, or if observations and distribution types are not compatible), mixedMemModel will throw an error. For additional details on usage, and model assumptions, see the corresponding vignette "Fitting Mixed Membership Models Using mixedMem".

Value
returns an object of class mixedMemModel
Examples

set.seed(123)
Total <- 50  # 50 Individuals
J <- 3  # 3 different variables
# distributions of each variable
dist <- c("multinomial", "bernoulli", "rank")
# 100 repeated measures of the multinomial, 5 repeated measures of the
# Bernoulli, 1 repeated measure of the rank
Rj <- c(100, 5, 1)

K <- 4  # 4 sub-populations
alpha <- rep(.5, K)  # hyperparameter for dirichlet distribution

# Number of categories/alternatives for each variable. For the Bernoulli, Vj = 1
Vj <- c(10, 1, 4)

theta <- array(0, dim = c(J, K, max(Vj)))
# Parameters governing multinomial
theta[1, ,] <- gtools::rdirichlet(K, rep(.3, Vj[1]))
# parameters governing Bernoulli
theta[2, ,] <- cbind(rbeta(K, 1, 1), matrix(0, nrow = K, ncol = Vj[1]-1))
theta[3, ,] <- cbind(gtools::rdirichlet(K, rep(.3, Vj[3])),
                matrix(0, nrow = K, ncol = Vj[1]-Vj[3]))

# Items selected for each observation. For Multinomial and Bernoulli, this is always 1
# For rank data, this will be the number of alternatives ranked
Nijr = array(0, dim = c(Total, J, max(Rj)))
Nijr[, 1, c(1:Rj[1])] = 1  # N_ijr is 1 for multinomial variables
Nijr[, 2, c(1:Rj[2])] = 1  # N_ijr is 1 for Bernoulli variables
Nijr[, 3, c(1:Rj[3])] = sample.int(Vj[3], size = Total, replace = TRUE)

# generate random sample of observations
sampleMixedMem <- rmixedMem(Total, J, Rj, Nijr, K, Vj, dist, theta, alpha)

## Initialize a mixedMemModel object
test_model <- mixedMemModel(Total = Total, J = J, Rj = Rj,
                            Nijr = Nijr, K = K, Vj = Vj, dist = dist, obs = sampleMixedMem$obs,
                            alpha = alpha, theta = theta)
# Look at Summary of the initialized model
summary(test_model)
# Plot the current values for theta
plot(test_model)
Description

`mmVarFit` is the primary function of the `mixedMem` package. The function fits parameters $\phi$ and $\delta$ for the variational distribution of latent variables as well as pseudo-MLE estimates for the population parameters $\alpha$ and $\theta$. See documentation for `mixedMemModel` or the package vignette, "Fitting Mixed Membership Models Using mixedMem" for a more detailed description of the variables and notation in a mixed membership model.

Usage

```r
mmVarFit(model, printStatus = 1, printMod = 1, stepType = 3,
maxTotalIter = 500, maxEIter = 1000, maxAlphaIter = 200,
maxThetaIter = 1000, maxLSIter = 400, elboTol = 1e-06,
alphaTol = 1e-06, thetaTol = 1e-10, aNaught = 1, tau = 0.899,
bMax = 3, bNaught = 1000, bMult = 1000, vCutoff = 13,
holdConst = c(-1))
```

Arguments

- `model`: a `mixedMemModel` object created by the `mixedMemModel` constructor.
- `printStatus`: an integer 0 or 1. When `printStatus` is 1 `mmVarFit` will print status updates, when `printStatus` is 0 `mmVarFit` will not print status updates.
- `printMod`: a positive integer which specifies how often to print status updates. The status will be printed at each step which is a multiple of `printMod`. The default value is 1.
- `stepType`: an integer from 0-3 which specifies what steps to carry out in estimation. 0 performs a single E-Step; this can be used to find the held out ELBO. 1 performs iterates between E-steps and M-steps for estimating theta but keeps alpha constant. 2 iterates between E-steps and M-steps for estimating alpha, but keeps theta constant. 3 completes full E and M steps. The default is 3.
- `maxTotalIter`: the maximum total steps before termination. A full E and M step together count as 1 step. If this maximum is ever achieved, a warning message will be printed at convergence. The default is 500.
- `maxEIter`: the maximum iterations for each E-Step. If this maximum is ever achieved, a warning message will be printed at convergence. The default is 1000.
- `maxAlphaIter`: the maximum iterations when fitting alpha. If this maximum is ever achieved, a warning message will be printed at convergence. The default is 200.
- `maxThetaIter`: the maximum iterations when fitting theta. If this maximum is ever achieved, a warning message will be printed at convergence. The default is 1000.
- `maxLSIter`: the maximum backtracking iterations in the line search for updating alpha and theta for rank data.
- `elboTol`: the convergence criteria for the EM Algorithm. When the relative increase in the ELBO is less than the convergence criteria, the algorithm converges. The default is 400.
- `alphaTol`: the convergence criteria for updates to alpha. When the relative increase in the ELBO is less than the convergence criteria, the update to alpha converges. The default is 1e-6.
**thetaTol** the convergence criteria for updates to theta. When the relative increase in the ELBO is less than the convergence criteria, the update to theta converges. The default is 1e-10.

**aNaught** the first step size in the backtracking line search used to update alpha and theta for rank data. The default is 1.

**tau** the backtracking factor in the backtracking line search used to update alpha and theta for rank data. The default is .899.

**bMax** the number of times the penalization factor is increased in the interior point method for fitting theta for rank data. The default is 3.

**bNaught** the initial penalization factor in the interior point method for fitting theta for rank data. bNaught must be positive and the default value is 1000.

**bMult** a positive number by which bNaught is multiplied by in each iteration of the interior point method for fitting theta for rank data. The default is 1000.

**vCutoff** a positive integer cutoff for Vj at which a gradient ascent method is used instead of the Newton Raphson interior point method. This is used to avoid inverting a large matrix. The default is 13.

**holdConst** a vector of integers specifying groups to be held fixed during the estimation procedure. The estimation algorithm will hold the theta parameters of these specific groups constant, but update all other parameters. The group numbers range from 0 to K - 1. To estimate all groups, use the default value of c(-1).

### Details

`mmVarFit` selects psuedo-MLE estimates for α and θ and approximates the posterior distribution for the latent variables through a mean field variational approach. The variational lower bound on the log-likelihood at the data given model parameters is given by Jensen’s inequality:

$$ E_Q[log[p(X, Z, \Lambda)] - E_Q[log[Q(Z, \Lambda|\phi, \delta)]] \leq \log P(\text{obs}|\alpha, \theta) $$

where

$$ Q = \prod_i [\text{Dirichlet}(\lambda_i|\phi_i) \prod_j^{R_i} \prod_r^{N_{i,j,r}} \text{Multinomial}(Z_{i,j,r,n}|\delta_{i,j,r,n})] $$

The left hand side of the above equation is often referred to as the Evidence Lower Bound (ELBO).

The global parameters α and θ are selected to maximize this bound as a psuedo-MLE procedure. In addition, it can be shown that maximizing the ELBO with respect to the variational parameters φ and δ is equivalent to minimizing the KL divergence between the tractable variational distribution and the true posterior.

The method uses a variational EM approach. The E-step considers the global parameters α and θ fixed and picks appropriate variational parameters φ and δ to minimize the KL divergence between the true posterior and the variational distribution (or maximize the ELBO). The M-step considers the variational parameters fixed and global parameters θ and α are selected which maximize the lower bound on the log-likelihood (ELBO).

### Value

`mmVarFit` returns a `mixedMemModel` object containing updated variational parameters and global parameters.
References


See Also

mixedMemModel

Examples

## The following example generates multivariate observations (both variables are multinomial)## from a mixed membership model. The observed data is then used to fit a## mixed membership model using mmVarFit

## Generate Data
Total <- 30 #30 Individuals
J <- 2 # 2 variables
dist <- rep("multinomial",J) # both variables are multinomial
Rj <- rep(100,J) # 100 repititions for each variable
# Nijr will always be 1 for multinomials and bernoulli's
Nijr <- array(1, dim = c(Total, J, max(Rj)))
K <- 4 # 4 sub-populations
alpha <- rep(0.5, K) # hyperparameter for dirichlet distribution
Vj <- rep(5, J) # each multinomial has 5 categories
theta <- array(0, dim = c(J, K, max(Vj)))
theta[1,,] <- gtools::rdirichlet(K, rep(0.3, 5))
theta[2,,] <- gtools::rdirichlet(K, rep(0.3, 5))
obs <- rmixedMem(Total, J, Rj, Nijr, K, Vj, dist, theta, alpha)$obs

## Initialize a mixedMemModel object
test_model <- mixedMemModel(Total = Total, J = J, Rj = Rj, Nijr = Nijr,
K = K, Vj = Vj, dist = dist, obs = obs,
alpha = alpha, theta = theta+0)

## Fit the mixed membership model
out <- mmVarFit(test_model)

---

permuteLabels

Mixed Membership Post-Processing

Description

Mixed Membership models are invariant to permutations of the sub-population labels; swapping the names of each sub-population yields an equivalent model. The ordering of the labels in a fitted model is dependent on the initialization points of the variational EM algorithm. The permuteLabels function returns a mixedMemModel object where the labels (for $\theta$, $\phi$, $\delta$ and $\alpha$) have been permuted according a given permutation of the integers 1 through $K$. The findLabels function can be used to find a permutation of the labels which most closely matches another fitted model.
plot.mixedMemModel

Usage

permuteLabels(model, perm)

Arguments

model a fitted mixedMemModel object which will be relabeled.
perm a vector of length K with integers 1:K. This is the permutation by which to relabel the mixedMemModel object such that group i in the returned mixedMemModel object corresponds to group perm[i] from the input mixedMemModel object.

Value

permuteLabels returns a mixedMemModel object such that group i in the returned mixedMemModel object corresponds to group perm[i] from the input mixedMemModel object.

See Also

findLabels

plot.mixedMemModel  Plot a Mixed Membership Model

Description

Generic S3 function to produce visual representation of a mixedMemModel object. This function calls either the vizTheta or the vizMem function.

Usage

## S3 method for class 'mixedMemModel'
plot(x, type = "theta", compare = NULL, main = NULL, varNames = NULL, groupNames = NULL, nrow = NULL, ncol = NULL, indices = NULL, fitNames = NULL, ...)

Arguments

x the mixedMemModel object to be plotted.
type a string which indicates which estimated parameters to plot; valid options are "theta" or "membership". vizTheta is called when the type is "theta" and vizMem is called when the type is "membership".
compare an array or matrix for comparison. When type = "theta", compare should be an array the same size as x$theta. When type = "membership", compare should be a matrix the same size as x$phi.
main the main figure title.
varNames a vector of strings corresponding to names for each variable if plot type is theta.
groupNames a vector of strings corresponding to labels for each sub-population.
nrow the number of rows for the grid of plots.
ncol the number of columns for the grid of plots. If plot type is "theta", this must be K, if plot type is "membership", this must be a positive integer.
indices a vector of integers. If the plot type is "membership", this indicates which individuals to plot. When plot type is "theta", this indicates which variables to plot.
fitNames a vector of strings corresponding to labels for each fit.
... additional parameters to be passed.

See Also
mixedMemModel, vizTheta, vizMem

### Description

Simulate data from a mixed membership model.

### Usage

```r
rmixedMem(Total, J, Rj, Nijr, K, Vj, dist, theta, alpha, lambda = NULL)
```

### Arguments

- **Total** the number of individuals in the sample.
- **J** the number of variables observed on each individual.
- **Rj** a vector of positive integers of length J specifying the number of repeated measurements for each variable.
- **Nijr** an array of dimension (Total, J, max(Rj)) indicating the number of ranking levels for each replication. For multinomial and Bernoulli variables, $N_{ijr[i,j,r]} = 1$. For rank variables, $N_{ijr[i,j,r]}$ indicates the number of items ranked for each individual.
- **K** the number of latent sub-populations.
- **Vj** a vector of length J specifying the number of possible candidates for each variable. For a Bernoulli variable $V_{j} = 1$. For a multinomial or rank variable, $V_{j}[j]$ is the number of possible categories/items.
- **dist** a vector of strings of length J specifying variable types. Options are "bernoulli", "multinomial" or "rank" corresponding to the distributions of the observed variables.
theta an 3 way array of dimensions (J,K,max(Vj)) which governs the variable distributions. Parameter theta[j,k,] governs the distribution of responses on variable j for an individually completely in sub-population k. If the number of items/categories differs across variables, any unused portions of theta should be set to 0.

alpha a positive K-length vector which is the parameter for the Dirichlet distribution of membership scores.

lambda an optional matrix of dimensions (Total, K) containing the membership scores for each individual. If the lambda argument is not specified, the group membership scores will be automatically sampled from a Dirichlet(alpha)

Details
rmixedMem simulates data from a mixed membership model given the specified parameters and dimensions. The function returns a random sample of observations obs, context indicators Z, and group membership scores lambda.

Value
rmixedMem returns a list containing a three items: A matrix of group memberships scores lambda, an array of context indicators Z and an array of observations obs.

summary.mixedMemModel  Summary of a Mixed Membership Model

Description
Generic S3 summary function for mixedMemModel class.

Usage
## S3 method for class 'mixedMemModel'
summary(object, ...)

Arguments

  object the mixedMemModel object to be summarized

  ... additional parameters

Details
summary provides a summary of the given mixedMemModel object. The function prints the ELBO, the dimensions of the model and each variable type.

See Also
mixedMemModel
Description

`vizMem` plots estimates for the group membership scores of each individual. This is the function called by the `mixedMemModel` class generic plot function.

Usage

```r
vizMem(model, compare = NULL, main = "Estimated Membership", nrow = NULL, ncol = NULL, indices = NULL, groupNames = NULL, fitNames = NULL)
```

Arguments

- `model`: the `mixedMemModel` object that will be plotted.
- `compare`: a matrix of the same dimensions as `model$phi` which contains parameters to compare against the fitted model.
- `main`: the main figure title.
- `nrow`: the number of rows in each plot.
- `ncol`: the number of columns in each plot.
- `indices`: the specific individuals which will be shown in the plot. If the argument is left blank, all individuals will be plotted. If the number of individuals to be plotted is larger than `nrow * ncol` then multiple plots will be produced.
- `groupNames`: a vector specifying labels for each sub-population.
- `fitNames`: a vector of length 2 containing strings which correspond to the names of the models (fitted and comparison).

Details

The estimates plotted are the normalized $\phi$, which are the posterior means from the variational distribution. The estimated group membership scores are shown in black, and the estimates from a comparison model (if available) are shown in red. Each plot represents an individual.

This is the function called by the plot generic function for `mixedMemModel` objects.
Description

vizTheta plots $\theta$, the parameters which govern the sub-population distributions of variables in a mixed membership model. The parameter $\theta_{j,k}$ specifies the distribution of variable $j$ for complete members of sub-population $k$. The estimated parameters from the given model are shown in black in each plot; the parameters from a comparison model (if available) are shown in red. Each row of plots represents a single variable, and each column of the plots represents a sub-population.

Usage

```r
vizTheta(model, compare = NULL, main = "Estimated Theta", varNames = NULL, groupNames = NULL, nrow = NULL, fitNames = NULL, indices = NULL)
```

Arguments

- `model`: the `mixedMemModel` object that will be plotted.
- `compare`: an array of the same dimensions as `model$theta` which contains values to compare against the fitted value.
- `main`: the main figure title.
- `varNames`: a vector of strings specifying labels for each variable.
- `groupNames`: a vector of strings specifying labels for each sub-population.
- `nrow`: the number of rows in each plot. If the argument is not specified, all variables will appear in one plot.
- `fitNames`: the names of the models plotted.
- `indices`: a vector which indicates specific variables to plot. If the argument is not specified, all variables will be plotted. If the number of variables to plot is greater than `nrow`, then multiple plots will be produced.

Details

This is the function called by the plot generic function for `mixedMemModel` objects.
Index

* grade of membership
  mixedMem-package, 2
* mixed membership
  mixedMem-package, 2
* variational inference
  mixedMem-package, 2

ANES, 4
computeBIC, 6
computeELBO, 7

findLabels, 7
gmv_theta, 9

mixedMem (mixedMem-package), 2
mixedMem-package, 2
mixedMemModel, 9
mmVarFit, 11

permuteLabels, 14
plot.mixedMemModel, 15

rmixedMem, 16

summary.mixedMemModel, 17

vizMem, 18
vizTheta, 19