

Package ‘PLMIX’

May 2, 2018

Type Package

Title Bayesian Analysis of Finite Mixtures of Plackett-Luce Models for Partial Rankings/Orderings

Version 2.0.1

Date 2018-05-02

Description Fit finite mixtures of Plackett-Luce models for partial top rankings/orderings within the Bayesian framework. It provides MAP point estimates via EM algorithm and posterior MCMC simulations via Gibbs Sampling. It also fits MLE as a special case of the noninformative Bayesian analysis with vague priors. In addition to inferential techniques, the package assists other fundamental phases of a model-based analysis for partial rankings/orderings, by including functions for data manipulation, simulation, descriptive summary, model selection and goodness-of-fit evaluation.

License GPL (>= 2)

Imports Rcpp (>= 0.12.16), abind, foreach, rccd, MCMCpack, gtools, label.switching, stats

LinkingTo Rcpp

RoxygenNote 6.0.1

Suggests doParallel

LazyData true

NeedsCompilation yes

Author Cristina Mollica [aut, cre],
Luca Tardella [aut]

Maintainer Cristina Mollica <cristina.mollica@uniroma1.it>

Repository CRAN

Date/Publication 2018-05-02 10:21:05 UTC

R topics documented:

| | |
|------------------|---|
| PLMIX-package | 2 |
| bicPLMIX | 4 |
| binary_group_ind | 5 |

| | |
|-------------------------------|----|
| d_apa | 6 |
| d_carconf | 7 |
| d_dublinwest | 8 |
| d_german | 9 |
| d_nascar | 10 |
| freq_to_unit | 11 |
| gibbsPLMIX | 12 |
| label_switchPLMIX | 14 |
| Loglikelihood | 16 |
| make_complete | 17 |
| make_partial | 18 |
| mapPLMIX | 20 |
| mapPLMIX_multistart | 22 |
| paired_comparisons | 24 |
| ppcheckPLMIX | 25 |
| ppcheckPLMIX_cond | 27 |
| rank_ord_switch | 29 |
| rank_summaries | 30 |
| rPLMIX | 31 |
| selectPLMIX | 32 |
| unit_to_freq | 35 |

| | |
|--------------|-----------|
| Index | 37 |
|--------------|-----------|

| | |
|---------------|--|
| PLMIX-package | <i>Bayesian Analysis of Finite Mixtures of Plackett-Luce Models for Partial Rankings/Orderings</i> |
|---------------|--|

Description

The PLMIX package for R provides functions to fit and analyze finite mixtures of Plackett-Luce models for partial top rankings/orderings within the Bayesian framework. It provides MAP point estimates via EM algorithm and posterior MCMC simulations via Gibbs Sampling. It also fits MLE as a special case of the noninformative Bayesian analysis with vague priors.

In addition to inferential techniques, the package assists other fundamental phases of a model-based analysis for partial rankings/orderings, by including functions for data manipulation, simulation, descriptive summary, model selection and goodness-of-fit evaluation.

Details

The Plackett-Luce model is one of the most popular and frequently applied parametric distributions to analyze partial top rankings/orderings of a finite set of items. The present package allows to account for unobserved sample heterogeneity of partially ranked data with a model-based analysis relying on Bayesian finite mixtures of Plackett-Luce models. The package provides a suite of functions that covers the fundamental phases of a model-based analysis:

Ranking data manipulation

[binary_group_ind](#) Binary group membership matrix from the mixture component labels.

`freq_to_unit` From the frequency distribution to the dataset of individual orderings/rankings.
`make_complete` Random completion of partial orderings/rankings data.
`make_partial` Censoring of complete orderings/rankings data.
`rank_ord_switch` From rankings to orderings and vice-versa.
`unit_to_freq` From the dataset of individual orderings/rankings to the frequency distribution.

Ranking data simulation

`rPLMIX` Random sample from a finite mixture of Plackett-Luce models.

Ranking data description

`paired_comparisons` Paired comparison frequencies.
`rank_summaries` Summary statistics of partial ranking/ordering data.

Model estimation

`label_switchPLMIX` Label switching adjustment for Bayesian mixtures of Plackett-Luce models.
`likPLMIX` Likelihood evaluation for a mixture of Plackett-Luce models.
`loglikPLMIX` Loglikelihood evaluation for a mixture of Plackett-Luce models.
`gibbsPLMIX` Bayesian analysis with MCMC posterior simulation via Gibbs sampling.
`mapPLMIX` MAP estimation via EM algorithm.
`mapPLMIX_multistart` MAP estimation via EM algorithm with multiple starting values.

Model selection

`bicPLMIX` BIC value for a mixture of Plackett-Luce models.
`selectPLMIX` Bayesian model selection criteria.

Model assessment

`ppcheckPLMIX` Posterior predictive diagnostics.
`ppcheckPLMIX_cond` Posterior predictive diagnostics conditionally on the number of ranked items.

Datasets

`d_apa` American Psychological Association Data (partial orderings).
`d_carconf` Car Configurator Data (partial orderings).
`d_dublinwest` Dublin West Data (partial orderings).
`d_german` German Sample Data (complete orderings).
`d_nascar` NASCAR Data (partial orderings).

Data have to be supplied as an object of class `matrix`, where missing positions/items are denoted with zero entries and `Rank = 1` indicates the most-liked alternative. For a more efficient implementation of the methods, partial sequences with a single missing entry should be preliminarily filled in, as they correspond to complete rankings/orderings. In the present setting, ties are not allowed. Some quantities frequently recalled in the manual are the following:

N Sample size.
 K Number of possible items.
 G Number of mixture components.
 L Size of the final posterior MCMC sample (after burn-in phase).

Author(s)

Cristina Mollica and Luca Tardella

Maintainer: Cristina Mollica <cristina.mollica@uniroma1.it>

References

Mollica, C. and Tardella, L. (2017). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika*, **82**(2), pages 442–458, ISSN: 0033-3123, <http://dx.doi.org/10.1007/s11336-016-9530-0>.

Mollica, C. and Tardella, L. (2014). Epitope profiling via mixture modeling for ranked data. *Statistics in Medicine*, **33**(21), pages 3738–3758, ISSN: 0277-6715, <http://onlinelibrary.wiley.com/doi/10.1002/sim.6224/full>.

bicPLMIX

BIC for a mixture of Plackett-Luce models

Description

Compute BIC value for a mixture of Plackett-Luce models fitted to partial orderings.

Usage

```
bicPLMIX(max_log_lik, pi_inv, G, ref_known = TRUE, ref_vary = FALSE)
```

Arguments

| | |
|-------------|--|
| max_log_lik | Maximized log-likelihood value. |
| pi_inv | Numeric $N \times K$ data matrix of partial orderings. |
| G | Number of mixture components. |
| ref_known | Logical: whether the component-specific reference orders are known (not to be estimated). Default is TRUE. |
| ref_vary | Logical: whether the reference orders vary across mixture components. Default is FALSE. |

Details

The `bicPLMIX` function allows to compute the BIC value from the output of alternative MLE methods for mixtures of Plackett-Luce models. The `max_log_lik` and the BIC values can be straightforwardly obtained from the output of the `mapPLMIX` and `mapPLMIX_multistart` functions when the default noninformative priors are adopted. The `ref_known` and `ref_vary` arguments accommodate for the more general mixture of Extended Plackett-Luce models (EPL), involving the additional reference order parameters (Mollica and Tardella 2014). Since the Plackett-Luce model is a special instance of the EPL with the reference order equal to the identity permutation $(1, \dots, K)$, the default values of `ref_known` and `ref_vary` are set equal, respectively, to TRUE and FALSE.

Value

A list of two named objects:

| | |
|-------------|---------------------------|
| max_log_lik | The max_log_lik argument. |
| bic | BIC value. |

Author(s)

Cristina Mollica and Luca Tardella

References

Mollica, C. and Tardella, L. (2017). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika*, **82**(2), pages 442–458, ISSN: 0033-3123, DOI: 10.1007/s11336-016-9530-0.

Mollica, C. and Tardella, L. (2014). Epitope profiling via mixture modeling for ranked data. *Statistics in Medicine*, **33**(21), pages 3738–3758, ISSN: 0277-6715, DOI: 10.1002/sim.6224.

Schwarz, G. (1978). Estimating the dimension of a model. *Ann. Statist.*, **6**(2), pages 461–464, ISSN: 0090-5364, DOI: 10.1002/sim.6224.

See Also

[mapPLMIX](#), [mapPLMIX_multistart](#)

Examples

```
data(d_carconf)
K <- ncol(d_carconf)
MAP_mult <- mapPLMIX_multistart(pi_inv=d_carconf, K=K, G=3, n_start=2, n_iter=400*3)
bicPLMIX(max_log_lik=MAP_mult$mod$max_objective, pi_inv=d_carconf, G=3)$bic

# Equivalently,
MAP_mult$mod$bic
```

| | |
|------------------|---------------------------------------|
| binary_group_ind | <i>Binary group membership matrix</i> |
|------------------|---------------------------------------|

Description

Construct the binary group membership matrix from the multinomial classification vector.

Usage

```
binary_group_ind(class, G)
```

Arguments

`class` Numeric vector of class memberships.
`G` Number of possible different classes.

Value

Numeric $\text{length}(\text{class}) \times G$ matrix of binary group memberships.

Author(s)

Cristina Mollica and Luca Tardella

Examples

```
binary_group_ind(c(3,1,5),6)
```

d_apa

American Psychological Association Data (partial orderings)

Description

The popular American Psychological Association dataset (`d_apa`) contains the results of the voting ballots of the 1980 presidential election. A total of $N = 15449$ voters ranked a maximum of $K = 5$ candidates, conventionally classified as research psychologists (candidate 1 and 3), clinical psychologists (candidate 4 and 5) and community psychologists (candidate 2). The winner of the election was candidate 3. The dataset is composed of partial top orderings of varying lengths. Missing positions are denoted with zero entries.

Usage

```
data(d_apa)
```

Format

Data matrix of partial orderings with $N = 15449$ rows and $K = 5$ columns, named as:

`rank1` Candidate occupying position 1 (most-liked).
`rank2` Candidate occupying position 2.
`rank3` Candidate occupying position 3.
`rank4` Candidate occupying position 4.
`rank6` Candidate occupying position 5 (least-liked).

References

- Mollica, C. and Tardella, L. (2017). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika*, **82**(2), pages 442–258, ISSN: 0033-3123, DOI: 10.1007/s11336-016-9530-0.
- Diaconis, P. W. (1988). Group representations in probability and statistics. *Lecture Notes-Monograph Series*, pages 94–96.
- Diaconis, P. W. (1987). Spectral analysis for ranked data. Technical Report 282, Dept of Statistics, Stanford University.

Examples

```
data(d_apa)
head(d_apa)

# Subset of complete sequences

d_apa_compl=d_apa[rowSums(d_apa!=0)>=(ncol(d_apa)-1),]
head(d_apa_compl)
```

d_carconf

Car Configurator Data (partial orderings)

Description

The Car Configurator dataset (d_carconf) came up from a marketing study aimed at investigating customer preferences toward different car features. A sample of $N = 435$ customers were asked to construct their car by using an online configurator system and to choose among $K = 6$ car modules in order of preference. The car features are labeled as: 1 = price, 2 = exterior design, 3 = brand, 4 = technical equipment, 5 = producing country and 6 = interior design. The survey did not require a complete ranking elicitation, therefore the dataset is composed of partial top orderings of varying lengths. Missing positions are denoted with zero entries.

Usage

```
data(d_carconf)
```

Format

Data matrix of partial orderings with $N = 435$ rows and $K = 6$ columns, named as:

- rank1 Item occupying position 1 (most-liked).
- rank2 Item occupying position 2.
- rank3 Item occupying position 3.
- rank4 Item occupying position 4.
- rank5 Item occupying position 5.
- rank6 Item occupying position 6 (least-liked).

References

- Mollica, C. and Tardella, L. (2017). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika*, **82**(2), pages 442–458, ISSN: 0033-3123, DOI: 10.1007/s11336-016-9530-0.
- Hatzinger, R. and Dittrich, R. (2012). Prefmod: An R package for modeling preferences based on paired comparisons, rankings, or ratings. *Journal of Statistical Software*, **48**(10), pages 1–31.
- Dabic, M. and Hatzinger, R. (2009). Zielgruppenadaequante Ablaeufe in Konfigurationssystemen - eine empirische Studie im Automobilmarkt - Partial Rankings. In Hatzinger, R., Dittrich, R. and Salzberger, T. (eds), *Praeferenzanalyse mit R: Anwendungen aus Marketing, Behavioural Finance und Human Resource Management*. Wien: Facultas.

Examples

```
data(d_carconf)
head(d_carconf)

# Subset of complete sequences

d_carconf_compl=d_carconf[rowSums(d_carconf!=0)>=(ncol(d_carconf)-1),]
head(d_carconf_compl)
```

d_dublinwest

Dublin West Data (partial orderings)

Description

The Dublin West dataset (`d_dublinwest`) contains the results of the voting ballots of the 2002 Irish general election from the Dublin West constituency. The Irish voting system allows voters to rank the candidates in order of preferences, rather than only specify the favorite one. In the Dublin West constituency, $N = 29988$ voters ranked a maximum of $K = 9$ candidates, labeled as: 1 = Bonnie R., 2 = Burton J., 3 = Doherty-Ryan D., 4 = Higgins J., 5 = Lenihan B., 6 = McDonald M., 7 = Morrissey T., 8 = Smyth J. and 9 = Terry S.. The dataset is composed of partial top orderings of varying lengths. Missing positions are denoted with zero entries.

Usage

```
data(d_dublinwest)
```

Format

Data matrix of partial orderings with $N = 29988$ rows and $K = 9$ columns. Each row lists the candidates from the most-liked (rank1) to the least-liked (rank9) in a given voting ballot.

Source

The 2002 Dublin West data have been downloaded from <http://www.preflib.org/> PrefLib: A Library for Preferences. In that repository, preferences with ties are also included. The original source was publicly available from the Dublin County Returning Officer at the following URL: <https://dublincountyreturningofficer.com/>.

References

- Gormley, I. C. and Murphy, T. B. (2008). Exploring Voting Blocs Within the Irish Electorate: A Mixture Modeling Approach. *Journal of the American Statistical Association*, **103**(483), pages 1014–1027.
- Gormley, I. C. and Murphy, T. B. (2009). A grade of membership model for rank data. *Bayesian Analysis*, **4**(2), pages 65–295.
- Mattei, N. and Walsh, T. (2013) PrefLib: A Library of Preference Data. *Proceedings of Third International Conference on Algorithmic Decision Theory (ADT 2013)*. Springer, Lecture Notes in Artificial Intelligence, November 13-15, 2013.

Examples

```
data(d_dublinwest)
head(d_dublinwest)

# Subset of complete sequences

d_dublinwest_compl=d_dublinwest[rowSums(d_dublinwest!=0)>=(ncol(d_dublinwest)-1),]
head(d_dublinwest_compl)
```

d_german

German Sample Data (complete orderings)

Description

The German Sample dataset (d_german) is part of a comparative cross-sectional study on political actions and mass participation involving five Western countries. The dataset regards a sample of $N = 2262$ German respondents who were asked to rank $K = 4$ political goals in order of desirability, namely: 1 = maintaining order in the nation, 2 = giving people more say in the decisions of government, 3 = fighting rising prices and 4 = protecting freedom of speech. The dataset is composed of complete orderings.

Usage

```
data(d_german)
```

Format

Data matrix of complete orderings with $N = 2262$ rows and $K = 4$ columns, named as:

rank1 Political goal occupying position 1 (most desirable).

rank2 Political goal occupying position 2.

rank3 Political goal occupying position 3.

rank4 Political goal occupying position 4 (least desirable).

References

Barnes, S. H. et al. (1979). Political action. Mass participation in five Western democracies. London: Sage.

Croon, M. A. (1989). Latent class models for the analysis of rankings. In De Soete, G., Feger, H. and Klauer, K. C. (eds), *New Developments in Psychological Choice Modeling*, pages 99–121. North-Holland: Amsterdam.

Examples

```
data(d_german)
head(d_german)
```

| | |
|----------|--|
| d_nascar | <i>NASCAR Data (partial orderings)</i> |
|----------|--|

Description

The NASCAR dataset (d_nascar) collects the results of the 2002 season of stock car racing held in the United States. The 2002 championship consisted of $N = 36$ races, with 43 car drivers competing in each race. A total of $K = 87$ drivers participated in the 2002 season, taking part to a different number of races: some of them competed in all the races, some others in only one. The results of the entire 2002 season were collected in the form of top-43 orderings, where the position of the not-competing drivers in each race is assumed lower than the 43th, but undetermined. Missing positions are denoted with zero entries.

Usage

```
data(d_nascar)
```

Format

Data matrix of partial orderings with $N = 36$ rows and $K = 87$ columns. Each row lists the car drivers from the top position (rank1) to the bottom one (rank87) in a given race. Columns from the 44th to the 87th are filled with zeros, because only 43 drivers competed in each race.

Source

The NASCAR dataset in the MATLAB format used by Hunter, D. R. (2004) can be downloaded from <http://sites.stat.psu.edu/~dhunter/code/btmatlab/>. At the same link, a .xls file with drivers' names is also available.

References

- Caron, F. and Doucet, A. (2012). Efficient Bayesian inference for Generalized Bradley-Terry models. *J. Comput. Graph. Statist.*, **21**(1), pages 174–196.
- Guiver, J. and Snelson, E. (2009). Bayesian inference for Plackett-Luce ranking models. In Bottou, L. and Littman, M., editors, *Proceedings of the 26th International Conference on Machine Learning - ICML 2009*, pages 377–384. Omnipress.
- Hunter, D. R. (2004). MM algorithms for Generalized Bradley-Terry models. *Ann. Statist.*, **32**(1), pages 384–406.

Examples

```
data(d_nascar)
head(d_nascar)

# Compute the number of races for each of the 87 drivers

table(c(d_nascar[,1:43]))

# Identify drivers arrived last (43th position) in all the races
# where they participated in

which(colSums(rank_summaries(d_nascar, format="ordering")$marginals[1:42,])==0)

# Obscure drivers 84, 85, 86 and 87 to get
# the reduced dataset with 83 racers employed by Hunter, D. R. (2004)

d_nascar_hunter=d_nascar[,1:83]
d_nascar_hunter[is.element(d_nascar_hunter,84:87)]=0
```

freq_to_unit

Individual rankings/orderings from the frequency distribution

Description

Construct the dataset of individual rankings/orderings from the frequency distribution of the distinct observed sequences.

Usage

```
freq_to_unit(freq_distr)
```

Arguments

freq_distr Numeric matrix of the distinct observed sequences with the corresponding frequencies indicated in the last ($K + 1$)-th column.

Value

Numeric $N \times K$ data matrix of observed individual sequences.

Author(s)

Cristina Mollica and Luca Tardella

Examples

```
library(gtools)

K <- 4
perm_matrix <- permutations(n=K,r=K)
freq_data <- cbind(perm_matrix,sample(1:factorial(K)))
freq_data

freq_to_unit(freq_distr=freq_data)
```

gibbsPLMIX

Gibbs sampling for a Bayesian mixture of Plackett-Luce models

Description

Perform Gibbs sampling simulation for a Bayesian mixture of Plackett-Luce models fitted to partial orderings.

Usage

```
gibbsPLMIX(pi_inv, K, G, init = list(z = NULL, p = NULL), n_iter = 1000,
  n_burn = 500, hyper = list(shape0 = matrix(1, nrow = G, ncol = K), rate0 =
  rep(0.001, G), alpha0 = rep(1, G)), centered_start = FALSE)
```

Arguments

| | |
|--------|--|
| pi_inv | Numeric $N \times K$ data matrix of partial orderings. |
| K | Number of possible items. |
| G | Number of mixture components. |
| init | List of named objects with initialization values: z is a numeric $N \times G$ matrix of binary component memberships; p is a numeric $G \times K$ matrix of component-specific support parameters. If starting values are not supplied (NULL), they are randomly generated with a uniform distribution. Default is NULL. |
| n_iter | Total number of MCMC iterations. |
| n_burn | Number of initial burn-in drawings removed from the final MCMC sample. |

| | |
|----------------|---|
| hyper | List of named objects with hyperparameter values for the conjugate prior specification: $\text{shape}\theta$ is a numeric $G \times K$ matrix of shape hyperparameters; $\text{rate}\theta$ is a numeric vector of G rate hyperparameters; $\text{alpha}\theta$ is a numeric vector of G Dirichlet hyperparameters. Default is vague prior setting. |
| centered_start | Logical: whether a random start whose support parameters and weights should be centered around the observed relative frequency that each item has been ranked top. Default is FALSE. Ignored when <code>init</code> is not NULL. |

Details

The size L of the final posterior sample is equal to `n_iter-n_burn`.

Value

A list of named objects:

| | |
|----------|---|
| W | Numeric $L \times G$ matrix with MCMC samples of the mixture weights. |
| P | Numeric $L \times (G * K)$ matrix with MCMC samples of the component-specific support parameters. |
| log_lik | Numeric vector of posterior log-likelihood values. |
| deviance | Numeric vector of posterior deviance values ($-2 * \text{log_lik}$). |

Author(s)

Cristina Mollica and Luca Tardella

References

- Mollica, C. and Tardella, L. (2017). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika*, **82**(2), pages 442–458, ISSN: 0033-3123, DOI: 10.1007/s11336-016-9530-0.
- Mollica, C. and Tardella, L. (2014). Epitope profiling via mixture modeling for ranked data. *Statistics in Medicine*, **33**(21), pages 3738–3758, ISSN: 0277-6715, DOI: 10.1002/sim.6224.

Examples

```
data(d_carconf)
GIBBS <- gibbsPLMIX(pi_inv=d_carconf, K=ncol(d_carconf), G=3, n_iter=30, n_burn=10)
str(GIBBS)

# Get posterior samples of Plackett-Luce mixture parameters
GIBBS$P
GIBBS$W
```

| | |
|-------------------|---|
| label_switchPLMIX | <i>Label switching adjustment for Bayesian mixtures of Plackett-Luce models</i> |
|-------------------|---|

Description

Remove the label switching phenomenon from the MCMC samples of Bayesian mixtures of Plackett-Luce models with a different number of components.

Usage

```
label_switchPLMIX(pi_inv, seq_G, MCMCsampleP, MCMCsampleW, MAPestP, MAPestW,
  parallel = FALSE)
```

Arguments

| | |
|-------------|---|
| pi_inv | Numeric $N \times K$ data matrix of partial orderings. |
| seq_G | Numeric vector with the number of components of the Plackett-Luce mixtures to be assessed. |
| MCMCsampleP | List of size <code>length(seq_G)</code> , whose generic element is a numeric $L \times (G * K)$ matrix with the MCMC samples of the component-specific support parameters to be processed. |
| MCMCsampleW | List of size <code>length(seq_G)</code> , whose generic element is a numeric $L \times G$ matrix with the MCMC samples of the mixture weights to be processed. |
| MAPestP | List of size <code>length(seq_G)</code> , whose generic element is a numeric $G \times K$ matrix with the MAP estimates of the component-specific support parameters to be used as pivot in the PRA method. |
| MAPestW | List of size <code>length(seq_G)</code> , whose generic element is a numeric vector with the MAP estimates of the G mixture weights to be used as pivot in the PRA method. |
| parallel | Logical: whether parallelization should be used. Default is FALSE. |

Details

The `label_switchPLMIX` function performs the label switching adjustment of the MCMC samples via the Pivotal Reordering Algorithm (PRA) described in Marin et al (2005), by recalling the `pra` function from the `label.switching` package.

Value

A list of named objects:

| | |
|---------------|---|
| final_sampleP | List of size <code>length(seq_G)</code> , whose generic element is a numeric $G \times K \times L$ array with the MCMC samples of the component-specific support parameters adjusted for label switching. |
| final_sampleW | List of size <code>length(seq_G)</code> , whose generic element is a numeric $L \times G$ matrix with the MCMC samples of the mixture weights adjusted for label switching. |

Author(s)

Cristina Mollica and Luca Tardella

References

- Mollica, C. and Tardella, L. (2017). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika*, **82**(2), pages 442–458, ISSN: 0033-3123, DOI: 10.1007/s11336-016-9530-0.
- Marin, J. M., Mengersen, K. and Robert, C.P. (2005). Bayesian modelling and inference on mixtures of distributions. Handbook of Statistics (25), D. Dey and C.R. Rao (eds). Elsevier-Sciences.

See Also

[pra](#)

Examples

```

data(d_carconf)

K <- ncol(d_carconf)
n.start <- 2

MAP_1 <- mapPLMIX_multistart(pi_inv=d_carconf, K=K, G=1,
                             n_start=n.start, n_iter=400*1)

MAP_2 <- mapPLMIX_multistart(pi_inv=d_carconf, K=K, G=2,
                             n_start=n.start, n_iter=400*2)

MAP_3 <- mapPLMIX_multistart(pi_inv=d_carconf, K=K, G=3,
                             n_start=n.start, n_iter=400*3)

mcmc_iter <- 30
burnin <- 10

GIBBS_1 <- gibbsPLMIX(pi_inv=d_carconf, K=K, G=1, n_iter=mcmc_iter,
                     n_burn=burnin, init=list(p=MAP_1$mod$P_map,
                                             z=binary_group_ind(MAP_1$mod$class_map,G=1)))
GIBBS_2 <- gibbsPLMIX(pi_inv=d_carconf, K=K, G=2, n_iter=mcmc_iter,
                     n_burn=burnin, init=list(p=MAP_2$mod$P_map,
                                             z=binary_group_ind(MAP_2$mod$class_map,G=2)))
GIBBS_3 <- gibbsPLMIX(pi_inv=d_carconf, K=K, G=3, n_iter=mcmc_iter,
                     n_burn=burnin, init=list(p=MAP_3$mod$P_map,
                                             z=binary_group_ind(MAP_3$mod$class_map,G=3)))

# Adjusting the MCMC samples for label switching

require(doParallel)
run_in_parallel <- !is.na(detectCores())
if(run_in_parallel){
  registerDoParallel(2)
  getDoParWorkers()
}

```

```

}
LS <- label_switchPLMIX(pi_inv=d_carconf, seq_G=1:3,
  MCMCsampleP=list(GIBBS_1$P, GIBBS_2$P, GIBBS_3$P),
  MCMCsampleW=list(GIBBS_1$W, GIBBS_2$W, GIBBS_3$W),
  MAPestP=list(MAP_1$mod$P_map, MAP_2$mod$P_map, MAP_3$mod$P_map),
  MAPestW=list(MAP_1$mod$W_map, MAP_2$mod$W_map, MAP_3$mod$W_map),
  parallel = run_in_parallel)

str(LS)

```

| | |
|---------------|---|
| Loglikelihood | <i>Likelihood and Log-likelihood evaluation for a mixture of Plackett-Luce models</i> |
|---------------|---|

Description

Compute either the likelihood or the log-likelihood of the Plackett-Luce mixture model parameters for a partial ordering dataset.

Usage

```
likPLMIX(p, ref_order, weights, pi_inv)
```

```
loglikPLMIX(p, ref_order, weights, pi_inv)
```

Arguments

| | |
|-----------|---|
| p | Numeric $G \times K$ matrix of component-specific support parameters. |
| ref_order | Numeric $G \times K$ matrix of component-specific reference orders. |
| weights | Numeric vector of G mixture weights. |
| pi_inv | Numeric $N \times K$ data matrix of partial orderings. |

Details

The `ref_order` argument accommodates for the more general mixture of Extended Plackett-Luce models (EPL), involving the additional reference order parameters (Mollica and Tardella 2014). A permutation of the first K integers can be specified in each row of the `ref_order` argument. Since the Plackett-Luce model is a special instance of the EPL with the reference order equal to the identity permutation, the `ref_order` argument must be a matrix with G rows equal to $(1, \dots, K)$ when dealing with Plackett-Luce mixtures.

Value

Either the likelihood or the log-likelihood value of the Plackett-Luce mixture model parameters for a partial ordering dataset.

Author(s)

Cristina Mollica and Luca Tardella

References

Mollica, C. and Tardella, L. (2017). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika*, **82**(2), pages 442–458, ISSN: 0033-3123, DOI: 10.1007/s11336-016-9530-0.

Mollica, C. and Tardella, L. (2014). Epitope profiling via mixture modeling for ranked data. *Statistics in Medicine*, **33**(21), pages 3738–3758, ISSN: 0277-6715, DOI: 10.1002/sim.6224.

Examples

```
data(d_apa)

K <- ncol(d_apa)
G <- 3
support_par <- matrix(1:(G*K), nrow=G, ncol=K)
weights_par <- c(0.50, 0.25, 0.25)

loglikPLMIX(p=support_par, ref_order=matrix(1:K, nrow=G, ncol=K, byrow=TRUE),
            weights=weights_par, pi_inv=d_apa)
```

make_complete

Completion of partial rankings/orderings

Description

Return complete rankings/orderings from partial sequences relying on a random generation of the missing positions/items.

Usage

```
make_complete(data, format = "ordering", nranked = NULL,
             probitems = rep(1, ncol(data)))
```

Arguments

| | |
|-----------|--|
| data | Numeric $N \times K$ data matrix of partial sequences to be completed. |
| format | Character string indicating the format of the data argument ("ordering" or "ranking"). |
| nranked | Optional numeric vector of length N with the number of items ranked by each sample unit. |
| probitems | Numeric vector with the K item-specific probabilities to be employed for the random generation of the missing positions/items (normalization is not necessary). See Details for further explanation. Default is equal probabilities. |

Details

The completion of the partial top rankings/orderings is performed according to the Plackett-Luce scheme, that is, with a sampling without replacement of the not-ranked items by using the positive values in the `probitems` argument as support parameters.

Value

A list of two named objects:

| | |
|---------------------------|---|
| <code>completedata</code> | Numeric $N \times K$ data matrix of complete sequences with the same format of the input data. |
| <code>nranked</code> | Numeric vector of length N with the number of items ranked by each sample unit of the input data. |

Author(s)

Cristina Mollica and Luca Tardella

References

Mollica, C. and Tardella, L. (2017). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika*, **82**(2), pages 442–458, ISSN: 0033-3123, DOI: 10.1007/s11336-016-9530-0.

Examples

```
# Completion based on the top item frequencies
data(d_dublinwest)
head(d_dublinwest)
top_item_freq <- rank_summaries(data=d_dublinwest, format="ordering", mean_rank=FALSE,
                               pc=FALSE)$marginals["Rank_1",]

d_dublinwest_compl <- make_complete(data=d_dublinwest, format="ordering",
                                   probitems=top_item_freq)
head(d_dublinwest_compl$completedata)
```

make_partial

Censoring of complete rankings/orderings

Description

Return partial top rankings/orderings from complete sequences obtained either with user-specified censoring patterns or with a random truncation.

Usage

```
make_partial(data, format = "ordering", nranked = NULL, probcens = rep(1,
                             ncol(data) - 1))
```

Arguments

| | |
|----------|--|
| data | Numeric $N \times K$ data matrix of complete sequences to be censored. |
| format | Character string indicating the format of the data argument ("ordering" or "ranking"). |
| nranked | Numeric vector of length N with the desired number of items ranked by each sample unit after censoring. If not supplied (NULL), the censoring patterns are randomly generated according to the probabilities in the probcens argument. |
| probcens | Numeric vector of length $(K - 1)$ with the probability of each censoring pattern to be employed for the random truncation of the complete sequences (normalization is not necessary). It works only if nranked argument is NULL. See Details for further explanation. Default is equal probabilities. |

Details

The censoring of the complete sequences can be performed in: (i) a deterministic way, by specifying the number of top positions to be retained for each sample unit in the nranked argument; (ii) a random way, by sequentially specifying the probabilities of the top-1, top-2, ..., top- $(K - 1)$ censoring patterns in the probcens argument. Recall that a top- $(K - 1)$ sequence corresponds to a complete ordering/ranking.

Value

A list of two named objects:

| | |
|-------------|--|
| partialdata | Numeric $N \times K$ data matrix of partial (censored) sequences with the same format of the input data and missing positions/items denoted with zero entries. |
| nranked | Numeric vector of length N with the number of items ranked by each sample unit after censoring. |

Author(s)

Cristina Mollica and Luca Tardella

References

Mollica, C. and Tardella, L. (2017). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika*, **82**(2), pages 442–458, ISSN: 0033-3123, DOI: 10.1007/s11336-016-9530-0.

Examples

```
data(d_german)
head(d_german)
d_german_cens <- make_partial(data=d_german, format="ordering",
                             probcens=c(0.3, 0.3, 0.4))
head(d_german_cens$partialdata)

# Check consistency with the nominal censoring probabilities
round(prop.table(table(d_german_cens$nranked)), 2)
```

mapPLMIX

*MAP estimation for a Bayesian mixture of Plackett-Luce models***Description**

Perform MAP estimation via EM algorithm for a Bayesian mixture of Plackett-Luce models fitted to partial orderings.

Usage

```
mapPLMIX(pi_inv, K, G, init = list(p = NULL, omega = NULL), n_iter = 1000,
  hyper = list(shape0 = matrix(1, nrow = G, ncol = K), rate0 = rep(0, G),
  alpha0 = rep(1, G)), eps = 10^(-6), centered_start = FALSE,
  plot_objective = TRUE)
```

Arguments

| | |
|----------------|---|
| pi_inv | Numeric $N \times K$ data matrix of partial orderings. |
| K | Number of possible items. |
| G | Number of mixture components. |
| init | List of named objects with initialization values: p is a numeric $G \times K$ matrix of component-specific support parameters; omega is a numeric vector of G mixture weights. If starting values are not supplied (NULL), they are randomly generated with a uniform distribution. Default is NULL. |
| n_iter | Maximum number of EM iterations. |
| hyper | List of named objects with hyperparameter values for the conjugate prior specification: shape0 is a numeric $G \times K$ matrix of shape hyperparameters; rate0 is a numeric vector of G rate hyperparameters; alpha0 is a numeric vector of G Dirichlet hyperparameters. Default is noninformative (flat) prior setting. |
| eps | Tolerance value for the convergence criterion. |
| centered_start | Logical: whether a random start whose support parameters and weights should be centered around the observed relative frequency that each item has been ranked top. Default is FALSE. Ignored when init is not NULL. |
| plot_objective | Logical: whether the objective function should be plotted. Default is FALSE. |

Details

Under noninformative (flat) prior setting, the EM algorithm for MAP estimation corresponds to the EMM algorithm described by Gormley and Murphy (2006) to perform frequentist inference. In this case, the MAP solution coincides with the MLE and the output vectors log_lik and objective coincide as well. The `mapPLMIX` function performs the MAP procedure with a single starting value. To address the issue of local maxima in the posterior distribution, see the `mapPLMIX_multistart` function.

Value

A list of named objects:

| | |
|---------------|--|
| P_map | Numeric $G \times K$ matrix with the MAP estimates of the component-specific support parameters. |
| W_map | Numeric vector with the MAP estimates of the G mixture weights. |
| z_hat | Numeric $N \times G$ matrix of estimated posterior component membership probabilities. |
| class_map | Numeric vector of N component memberships based on MAP allocation from the z_hat matrix. |
| log_lik | Numeric vector of log-likelihood values at each iteration. |
| objective | Numeric vector of objective function values at each iteration. |
| max_objective | Maximized objective function value. |
| bic | BIC value (only for the default flat priors, otherwise NULL). |
| conv | Binary convergence indicator: 1 = convergence has been achieved, 0 = otherwise. |

Author(s)

Cristina Mollica and Luca Tardella

References

- Mollica, C. and Tardella, L. (2017). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika*, **82**(2), pages 442–458, ISSN: 0033-3123, DOI: 10.1007/s11336-016-9530-0.
- Mollica, C. and Tardella, L. (2014). Epitope profiling via mixture modeling for ranked data. *Statistics in Medicine*, **33**(21), pages 3738–3758, ISSN: 0277-6715, DOI: 10.1002/sim.6224.
- Gormley, I. C. and Murphy, T. B. (2006). Analysis of Irish third-level college applications data. *Journal of the Royal Statistical Society: Series A*, **169**(2), pages 361–379, ISSN: 0964-1998, DOI: 10.1111/j.1467-985X.2006.00412.x.

See Also

[mapPLMIX_multistart](#)

Examples

```
data(d_carconf)

MAP <- mapPLMIX(pi_inv=d_carconf, K=ncol(d_carconf), G=3, n_iter=400*3)
str(MAP)
MAP$P_map
MAP$W_map
```

mapPLMIX_multistart *MAP estimation for a Bayesian mixture of Plackett-Luce models with multiple starting values*

Description

Perform MAP estimation via EM algorithm with multiple starting values for a Bayesian mixture of Plackett-Luce models fitted to partial orderings.

Usage

```
mapPLMIX_multistart(pi_inv, K, G, n_start = 1, init = rep(list(list(p =
  NULL, omega = NULL)), times = n_start), n_iter = 200, hyper = list(shape0
  = matrix(1, nrow = G, ncol = K), rate0 = rep(0, G), alpha0 = rep(1, G)),
  eps = 10^(-6), plot_objective = FALSE, init_index = 1:n_start,
  parallel = FALSE, centered_start = FALSE)
```

Arguments

| | |
|----------------|--|
| pi_inv | Numeric $N \times K$ data matrix of partial orderings. |
| K | Number of possible items. |
| G | Number of mixture components. |
| n_start | Number of starting values. |
| init | List of <code>n_start</code> lists of named objects with initialization values: <code>p</code> is a numeric $G \times K$ matrix of component-specific support parameters; <code>omega</code> is a numeric vector of G mixture weights. If starting values are not supplied (NULL), they are randomly generated with a uniform distribution. Default is NULL. |
| n_iter | Maximum number of EM iterations. |
| hyper | List of named objects with hyperparameter values for the conjugate prior specification: <code>shape0</code> is a numeric $G \times K$ matrix of shape hyperparameters; <code>rate0</code> is a numeric vector of G rate hyperparameters; <code>alpha0</code> is a numeric vector of G Dirichlet hyperparameters. Default is noninformative (flat) prior setting. |
| eps | Tolerance value for the convergence criterion. |
| plot_objective | Logical: whether the objective function should be plotted. Default is FALSE. |
| init_index | Numeric vector indicating the positions of the starting values in the <code>init</code> list to be actually launched. Useful to launch the most promising starting values identified after a preliminary run. Default is run all the starting points in the <code>init</code> list. |
| parallel | Logical: whether parallelization should be used. Default is FALSE. |
| centered_start | Logical: whether a random start whose support parameters and weights should be centered around the observed relative frequency that each item has been ranked top. Default is FALSE. Ignored when <code>init</code> is not NULL. |

Details

Under noninformative (flat) prior setting, the EM algorithm for MAP estimation corresponds to the EMM algorithm described by Gormley and Murphy (2006) to perform frequentist inference. In this case the MAP solution coincides with the MLE. The best model in terms of maximized posterior distribution is returned.

Value

A list of named objects:

| | |
|---------------|---|
| mod | List of named objects describing the best model in terms of maximized posterior distribution. See output values of the single-run <code>mapPLMIX</code> function for a detailed explanation of the list elements. |
| max_objective | Numeric vector of the maximized objective function values for each initialization. |
| convergence | Binary vector with <code>length(init_index)</code> convergence indicators for each initialization: 1 = convergence has been achieved, 0 = otherwise. |

Author(s)

Cristina Mollica and Luca Tardella

References

- Mollica, C. and Tardella, L. (2017). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika*, **82**(2), pages 442–458, ISSN: 0033-3123, DOI: 10.1007/s11336-016-9530-0.
- Mollica, C. and Tardella, L. (2014). Epitope profiling via mixture modeling for ranked data. *Statistics in Medicine*, **33**(21), pages 3738–3758, ISSN: 0277-6715, DOI: 10.1002/sim.6224.
- Gormley, I. C. and Murphy, T. B. (2006). Analysis of Irish third-level college applications data. *Journal of the Royal Statistical Society: Series A*, **169**(2), pages 361–379, ISSN: 0964-1998, DOI: 10.1111/j.1467-985X.2006.00412.x.

See Also

[mapPLMIX](#)

Examples

```
require(MCMCpack)

data(d_carconf)

MAP_mult <- mapPLMIX_multistart(pi_inv=d_carconf, K=ncol(d_carconf), G=3,
                               n_start=2, n_iter=400*3)

str(MAP_mult)
MAP_mult$mod$P_map
MAP_mult$mod$W_map
```

paired_comparisons *Paired comparison matrix for a partial ordering/ranking dataset*

Description

Construct the paired comparison matrix for a partial ordering/ranking dataset.

Usage

```
paired_comparisons(data, format = "ordering", nranked = NULL)
```

Arguments

| | |
|---------|--|
| data | Numeric $N \times K$ data matrix of partial sequences. |
| format | Character string indicating the format of the data argument ("ordering" or "ranking"). |
| nranked | Optional numeric vector of length N with the number of items ranked by each sample unit. |

Value

Numeric $K \times K$ paired comparison matrix: the (i, i') -th entry indicates the number of sample units that preferred item i to item i' .

Author(s)

Cristina Mollica and Luca Tardella

References

Mollica, C. and Tardella, L. (2017). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika*, **82**(2), pages 442–458, ISSN: 0033-3123, DOI: 10.1007/s11336-016-9530-0.

See Also

[rank_summaries](#)

Examples

```
data(d_dublinwest)
paired_comparisons(data=d_dublinwest, format="ordering")
```

| | |
|--------------|---|
| ppcheckPLMIX | <i>Posterior predictive check for Bayesian mixtures of Plackett-Luce models</i> |
|--------------|---|

Description

Perform posterior predictive check to assess the goodness-of-fit of Bayesian mixtures of Plackett-Luce models with a different number of components.

Usage

```
ppcheckPLMIX(pi_inv, seq_G, MCMCsampleP, MCMCsampleW, top1 = TRUE,
             paired = TRUE, parallel = FALSE)
```

Arguments

| | |
|-------------|---|
| pi_inv | Numeric $N \times K$ data matrix of partial orderings. |
| seq_G | Numeric vector with the number of components of the Plackett-Luce mixtures to be assessed. |
| MCMCsampleP | List of size $\text{length}(\text{seq}_G)$, whose generic element is a numeric $L \times (G * K)$ matrix with the MCMC samples of the component-specific support parameters. |
| MCMCsampleW | List of size $\text{length}(\text{seq}_G)$, whose generic element is a numeric $L \times G$ matrix with the MCMC samples of the mixture weights. |
| top1 | Logical: whether the posterior predictive p -value based on the top item frequencies has to be computed. Default is TRUE. |
| paired | Logical: whether the posterior predictive p -value based on the paired comparison frequencies has to be computed. Default is TRUE. |
| parallel | Logical: whether parallelization should be used. Default is FALSE. |

Details

The ppcheckPLMIX function returns two posterior predictive p -values based on two chi squared discrepancy variables involving: (i) the top item frequencies and (ii) the paired comparison frequencies. In the presence of partial sequences in the pi_inv matrix, the same missingness patterns observed in the dataset (i.e., the number of items ranked by each sample unit) are reproduced on the replicated datasets from the posterior predictive distribution.

Value

A list of named objects:

| | |
|------------------|--|
| post_pred_pvalue | Numeric $\text{length}(\text{seq}_G) \times 2$ matrix of posterior predictive p -values based on the top item and paired comparison frequencies. If top1 or paired argument is FALSE, the corresponding matrix entries are NA. |
|------------------|--|

Author(s)

Cristina Mollica and Luca Tardella

References

Mollica, C. and Tardella, L. (2017). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika*, **82**(2), pages 442–458, ISSN: 0033-3123, DOI: 10.1007/s11336-016-9530-0.

See Also

[ppcheckPLMIX_cond](#)

Examples

```

data(d_carconf)

K <- ncol(d_carconf)
n.start <- 2

MAP_1 <- mapPLMIX_multistart(pi_inv=d_carconf, K=K, G=1,
                             n_start=n.start, n_iter=400*1)

MAP_2 <- mapPLMIX_multistart(pi_inv=d_carconf, K=K, G=2,
                             n_start=n.start, n_iter=400*2)

MAP_3 <- mapPLMIX_multistart(pi_inv=d_carconf, K=K, G=3,
                             n_start=n.start, n_iter=400*3)

mcmc_iter <- 30
burnin <- 10

GIBBS_1 <- gibbsPLMIX(pi_inv=d_carconf, K=K, G=1, n_iter=mcmc_iter,
                     n_burn=burnin, init=list(p=MAP_1$mod$P_map,
                                               z=binary_group_ind(MAP_1$mod$class_map,G=1)))
GIBBS_2 <- gibbsPLMIX(pi_inv=d_carconf, K=K, G=2, n_iter=mcmc_iter,
                     n_burn=burnin, init=list(p=MAP_2$mod$P_map,
                                               z=binary_group_ind(MAP_2$mod$class_map,G=2)))
GIBBS_3 <- gibbsPLMIX(pi_inv=d_carconf, K=K, G=3, n_iter=mcmc_iter,
                     n_burn=burnin, init=list(p=MAP_3$mod$P_map,
                                               z=binary_group_ind(MAP_3$mod$class_map,G=3)))

CHECK <- ppcheckPLMIX(pi_inv=d_carconf, seq_G=1:3,
                     MCMCsampleP=list(GIBBS_1$P, GIBBS_2$P, GIBBS_3$P),
                     MCMCsampleW=list(GIBBS_1$W, GIBBS_2$W, GIBBS_3$W))
CHECK$post_pred_pvalue

```

| | |
|-------------------|---|
| ppcheckPLMIX_cond | <i>Conditional posterior predictive check for Bayesian mixtures of Plackett-Luce models</i> |
|-------------------|---|

Description

Perform conditional posterior predictive check to assess the goodness-of-fit of Bayesian mixtures of Plackett-Luce models with a different number of components.

Usage

```
ppcheckPLMIX_cond(pi_inv, seq_G, MCMCsampleP, MCMCsampleW, top1 = TRUE,
  paired = TRUE, parallel = FALSE)
```

Arguments

| | |
|-------------|---|
| pi_inv | Numeric $N \times K$ data matrix of partial orderings. |
| seq_G | Numeric vector with the number of components of the Plackett-Luce mixtures to be assessed. |
| MCMCsampleP | List of size $\text{length}(\text{seq}_G)$, whose generic element is a numeric $L \times (G * K)$ matrix with the MCMC samples of the component-specific support parameters. |
| MCMCsampleW | List of size $\text{length}(\text{seq}_G)$, whose generic element is a numeric $L \times G$ matrix with the MCMC samples of the mixture weights. |
| top1 | Logical: whether the posterior predictive p -value based on the top item frequencies has to be computed. Default is TRUE. |
| paired | Logical: whether the posterior predictive p -value based on the paired comparison frequencies has to be computed. Default is TRUE. |
| parallel | Logical: whether parallelization should be used. Default is FALSE. |

Details

The ppcheckPLMIX_cond function returns two posterior predictive p -values based on two chi squared discrepancy variables involving: (i) the top item frequencies and (ii) the paired comparison frequencies. In the presence of partial sequences in the pi_inv matrix, the same missingness patterns observed in the dataset (i.e., the number of items ranked by each sample unit) are reproduced on the replicated datasets from the posterior predictive distribution. Differently from the ppcheckPLMIX function, the conditional discrepancy measures are obtained by summing up the chi squared discrepancies computed on subsamples of observations with the same number of ranked items.

Value

A list of named objects:

| | |
|-----------------------|--|
| post_pred_pvalue_cond | Numeric $\text{length}(\text{seq}_G) \times 2$ matrix of posterior predictive p -values based on the top item and paired comparison frequencies. If top1 or paired argument is FALSE, the corresponding matrix entries are NA. |
|-----------------------|--|

Author(s)

Cristina Mollica and Luca Tardella

References

Mollica, C. and Tardella, L. (2017). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika*, **82**(2), pages 442–458, ISSN: 0033-3123, DOI: 10.1007/s11336-016-9530-0.

See Also

[ppcheckPLMIX](#)

Examples

```

data(d_carconf)

K <- ncol(d_carconf)
n.start <- 2

MAP_1 <- mapPLMIX_multistart(pi_inv=d_carconf, K=K, G=1,
                             n_start=n.start, n_iter=400*1)

MAP_2 <- mapPLMIX_multistart(pi_inv=d_carconf, K=K, G=2,
                             n_start=n.start, n_iter=400*2)

MAP_3 <- mapPLMIX_multistart(pi_inv=d_carconf, K=K, G=3,
                             n_start=n.start, n_iter=400*3)

mcmc_iter <- 30
burnin <- 10

GIBBS_1 <- gibbsPLMIX(pi_inv=d_carconf, K=K, G=1, n_iter=mcmc_iter,
                     n_burn=burnin, init=list(p=MAP_1$mod$P_map,
                                               z=binary_group_ind(MAP_1$mod$class_map,G=1)))
GIBBS_2 <- gibbsPLMIX(pi_inv=d_carconf, K=K, G=2, n_iter=mcmc_iter,
                     n_burn=burnin, init=list(p=MAP_2$mod$P_map,
                                               z=binary_group_ind(MAP_2$mod$class_map,G=2)))
GIBBS_3 <- gibbsPLMIX(pi_inv=d_carconf, K=K, G=3, n_iter=mcmc_iter,
                     n_burn=burnin, init=list(p=MAP_3$mod$P_map,
                                               z=binary_group_ind(MAP_3$mod$class_map,G=3)))

CHECKCOND <- ppcheckPLMIX_cond(pi_inv=d_carconf, seq_G=1:3,
                              MCMCsampleP=list(GIBBS_1$P, GIBBS_2$P, GIBBS_3$P),
                              MCMCsampleW=list(GIBBS_1$W, GIBBS_2$W, GIBBS_3$W))

CHECKCOND$post_pred_pvalue

```

| | |
|-----------------|---|
| rank_ord_switch | <i>Switch from orderings to rankings and vice versa</i> |
|-----------------|---|

Description

Convert the format of the input dataset from orderings to rankings and vice versa.

Usage

```
rank_ord_switch(data, format = "ordering", nranked = NULL)
```

Arguments

| | |
|---------|--|
| data | Numeric $N \times K$ data matrix of partial sequences whose format has to be converted. |
| format | Character string indicating the format of the data argument ("ordering" or "ranking"). |
| nranked | Optional numeric vector of length N with the number of items ranked by each sample unit. |

Value

Numeric $N \times K$ data matrix of partial sequences with inverse format.

Author(s)

Cristina Mollica and Luca Tardella

References

Mollica, C. and Tardella, L. (2017). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika*, **82**(2), pages 442–458, ISSN: 0033-3123, DOI: 10.1007/s11336-016-9530-0.

Mollica, C. and Tardella, L. (2014). Epitope profiling via mixture modeling for ranked data. *Statistics in Medicine*, **33**(21), pages 3738–3758, ISSN: 0277-6715, DOI: 10.1002/sim.6224.

Examples

```
# From orderings to rankings for the Dublin West dataset
data(d_dublinwest)
head(d_dublinwest)
rank_ord_switch(data=head(d_dublinwest), format="ordering")
```

rank_summaries *Descriptive summaries for a partial ordering/ranking dataset*

Description

Compute rank summaries and censoring patterns for a partial ordering/ranking dataset.

Usage

```
rank_summaries(data, format = "ordering", mean_rank = TRUE,
               marginals = TRUE, pc = TRUE)
```

Arguments

| | |
|-----------|--|
| data | Numeric $N \times K$ data matrix of partial sequences. |
| format | Character string indicating the format of the data argument ("ordering" or "ranking"). |
| mean_rank | Logical: whether the mean rank vector has to be computed. Default is TRUE. |
| marginals | Logical: whether the marginal rank distributions have to be computed. Default is TRUE. |
| pc | Logical: whether the paired comparison matrix has to be computed. Default is TRUE. |

Value

A list of named objects:

| | |
|---------------|---|
| nranked | Numeric vector of length N with the number of items ranked by each sample unit. |
| nranked_distr | Frequency distribution of the nranked vector. |
| na_or_not | Numeric $3 \times K$ matrix with the counts of sample units that ranked or not each item. The last row contains the total by column, corresponding to the sample size N . |
| mean_rank | Numeric vector of length K with the mean rank of each item. |
| marginals | Numeric $K \times K$ matrix of the marginal rank distributions: the (i, j) -th entry indicates the number of units that ranked item i in the j -th position. |
| pc | Numeric $K \times K$ paired comparison matrix: the (i, i') -th entry indicates the number of sample units that preferred item i to item i' . |

Author(s)

Cristina Mollica and Luca Tardella

References

Mollica, C. and Tardella, L. (2017). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika*, **82**(2), pages 442–458, ISSN: 0033-3123, DOI: 10.1007/s11336-016-9530-0.

Examples

```
data(d_carconf)
rank_summaries(data=d_carconf, format="ordering")
```

rPLMIX

Random sample from a mixture of Plackett-Luce models

Description

Draw a random sample of complete orderings/rankings from a G -component mixture of Plackett-Luce models.

Usage

```
rPLMIX(n = 1, K, G, p = t(matrix(1/K, nrow = K, ncol = G)),
  ref_order = t(matrix(1:K, nrow = K, ncol = G)), weights = rep(1/G, G),
  format = "ordering")
```

Arguments

| | |
|-----------|--|
| n | Number of observations to be sampled. Default is 1. |
| K | Number of possible items. |
| G | Number of mixture components. |
| p | Numeric $G \times K$ matrix of component-specific support parameters. Default is equal support parameters (uniform mixture components). |
| ref_order | Numeric $G \times K$ matrix of component-specific reference orders. Default is forward orders (identity permutations) in each row, corresponding to Plackett-Luce mixture components. See details for further explanation. |
| weights | Numeric vector of G mixture weights. Default is equal weights. |
| format | Character string indicating the format of the simulated dataset. |

Details

Positive values are required for p and $weights$ arguments (normalization is not necessary). The ref_order argument accommodates for the more general mixture of Extended Plackett-Luce models (EPL), involving the additional reference order parameters (Mollica and Tardella 2014). A permutation of the first K integers can be specified in each row of the ref_order argument to generate a sample from a G -component mixture of EPL. Since the Plackett-Luce model is a special instance of the EPL with the reference order equal to the identity permutation $(1, \dots, K)$, the default value of the ref_order argument is forward orders.

Value

If $G = 1$, a numeric $N \times K$ matrix of simulated complete sequences. If $G > 1$, a list of two named objects:

comp Numeric vector of N component memberships.
 sim_data Numeric $N \times K$ matrix of simulated complete sequences.

Author(s)

Cristina Mollica and Luca Tardella

References

Mollica, C. and Tardella, L. (2017). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika*, **82**(2), pages 442–458, ISSN: 0033-3123, DOI: 10.1007/s11336-016-9530-0.

Mollica, C. and Tardella, L. (2014). Epitope profiling via mixture modeling for ranked data. *Statistics in Medicine*, **33**(21), pages 3738–3758, ISSN: 0277-6715, DOI: 10.1002/sim.6224.

Examples

```
K <- 6
G <- 3
support_par <- matrix(1:(G*K), nrow=G, ncol=K)
weights_par <- c(0.50, 0.25, 0.25)

set.seed(47201)
simulated_data <- rPLMIX(n=5, K=K, G=G, p=support_par, weights=weights_par)
simulated_data$comp
simulated_data$sim_data
```

 selectPLMIX

Bayesian selection criteria for mixtures of Plackett-Luce models

Description

Compute Bayesian comparison criteria for mixtures of Plackett-Luce models with a different number of components.

Usage

```
selectPLMIX(pi_inv, seq_G, MCMCsampleP = vector(mode = "list", length =
  length(seq_G)), MCMCsampleW = vector(mode = "list", length = length(seq_G)),
  MAPestP, MAPestW, deviance, post_summary = "mean", parallel = FALSE)
```


Arguments

| | |
|--------------|---|
| pi_inv | Numeric $N \times K$ data matrix of partial orderings. |
| seq_G | Numeric vector with the number of components of the Plackett-Luce mixtures to be compared. |
| MCMCsampLeP | List of size $\text{length}(\text{seq}_G)$, whose generic element is a numeric $L \times (G * K)$ matrix with the MCMC samples of the component-specific support parameters. Default is list of NULL elements. |
| MCMCsampLeW | List of size $\text{length}(\text{seq}_G)$, whose generic element is a numeric $L \times G$ matrix with the MCMC samples of the mixture weights. Default is list of NULL elements. |
| MAPestP | List of size $\text{length}(\text{seq}_G)$, whose generic element is a numeric $G \times K$ matrix with the MAP estimates of the component-specific support parameters. |
| MAPestW | List of size $\text{length}(\text{seq}_G)$, whose generic element is a numeric vector with the MAP estimates of the G mixture weights. |
| deviance | List of size $\text{length}(\text{seq}_G)$, whose generic element is a numeric vector of posterior deviance values. |
| post_summary | Character string indicating the summary statistic for computing the point estimates of the Plackett-Luce mixture parameters from the MCMC sample. This argument is ignored when MAP estimates are supplied in the MAPestP and MAPestW arguments. Default is "mean". Alternatively one can choose "median". See details for further explanation. |
| parallel | Logical: whether parallelization should be used. Default is FALSE. |

Details

The `selectPLMIX` function privileges the use of the MAP point estimates to compute the Bayesian model comparison criteria, since they are not affected by the label switching issue. By setting both the `MAPestP` and `MAPestW` arguments equal to `NULL`, the user can alternatively compute the selection measures by relying on a different posterior summary ("mean" or "median") specified in the `post_summary` argument. In the latter case, the MCMC samples for each Plackett-Luce mixture must be supplied in the lists `MCMCsampLeP` and `MCMCsampLeW`. The drawback when working with point estimates other than the MAP is that the possible presence of label switching has to be previously removed from the traces to obtain meaningful results. See [ppcheckPLMIX](#) and [ppcheckPLMIX_cond](#) functions to perform label switching adjustment.

Several model selection criteria are returned. The two versions of DIC correspond to alternative ways of computing the effective number of parameters: DIC1 was proposed by Spiegelhalter et al. (2002) with penalty named p_D , whereas DIC2 was proposed by Gelman et al. (2004) with penalty named p_V . The latter coincides with the AICM introduced by Raftery et al. (2007), that is, the Bayesian counterpart of AIC. BPIC1 and BPIC2 are obtained from the two DIC by simply doubling the penalty term, as suggested by Ando (2007) to contrast DIC's tendency to overfitting. BICM1 is the Bayesian variant of the BIC, originally presented by Raftery et al. (2007) and entirely based on the MCMC sample. The BICM2, instead, involved the MAP estimate without the need of its approximation from the MCMC sample as for the BICM1.


```

MAP_2 <- mapPLMIX_multistart(pi_inv=d_carconf, K=K, G=2,
                             n_start=n.start, n_iter=400*2)

mcmc_iter <- 30
burnin <- 10

GIBBS_1 <- gibbsPLMIX(pi_inv=d_carconf, K=K, G=1, n_iter=mcmc_iter,
                     n_burn=burnin, init=list(p=MAP_1$mod$P_map,
                                             z=binary_group_ind(MAP_1$mod$class_map,G=1)))
GIBBS_2 <- gibbsPLMIX(pi_inv=d_carconf, K=K, G=2, n_iter=mcmc_iter,
                     n_burn=burnin, init=list(p=MAP_2$mod$P_map,
                                             z=binary_group_ind(MAP_2$mod$class_map,G=2)))

SELECT <- selectPLMIX(pi_inv=d_carconf, seq_G=1:2,
                     MAPestP=list(MAP_1$mod$P_map, MAP_2$mod$P_map),
                     MAPestW=list(MAP_1$mod$W_map, MAP_2$mod$W_map),
                     deviance=list(GIBBS_1$deviance, GIBBS_2$deviance))

SELECT$criteria

```

unit_to_freq

Frequency distribution from the individual rankings/orderings.

Description

Construct the frequency distribution of the distinct observed sequences from the dataset of individual rankings/orderings.

Usage

```
unit_to_freq(data)
```

Arguments

data Numeric $N \times K$ data matrix of observed individual sequences.

Value

Numeric matrix of the distinct observed sequences with the corresponding frequencies indicated in the last $(K + 1)$ -th column.

Author(s)

Cristina Mollica and Luca Tardella

Examples

```
# Frequency distribution of complete orderings
data(d_german)
unit_to_freq(data=d_german)

# Frequency distribution of partial orderings
data(d_apa)
unit_to_freq(data=d_apa)
```

Index

*Topic **Ranking data, Plackett-Luce model, Mixture models, Bayesian inference, MAP estimation, MCMC approximation**

PLMIX-package, 2

*Topic **datasets**

d_apa, 6

d_carconf, 7

d_dublinwest, 8

d_german, 9

d_nascar, 10

*Topic

PLMIX-package, 2

bicPLMIX, 3, 4

binary_group_ind, 2, 5

d_apa, 3, 6

d_carconf, 3, 7

d_dublinwest, 3, 8

d_german, 3, 9

d_nascar, 3, 10

freq_to_unit, 3, 11

gibbsPLMIX, 3, 12

label_switching, 14

label_switchPLMIX, 3, 14

Likelihood (Loglikelihood), 16

likelihood (Loglikelihood), 16

likPLMIX, 3

likPLMIX (Loglikelihood), 16

Loglikelihood, 16

loglikelihood (Loglikelihood), 16

loglikPLMIX, 3

loglikPLMIX (Loglikelihood), 16

make_complete, 3, 17

make_partial, 3, 18

mapPLMIX, 3–5, 20, 20, 23

mapPLMIX_multistart, 3–5, 20, 21, 22

paired_comparisons, 3, 24

PLMIX (PLMIX-package), 2

PLMIX-package, 2

ppcheckPLMIX, 3, 25, 28, 33

ppcheckPLMIX_cond, 3, 26, 27, 33

pra, 14, 15

rank_ord_switch, 3, 29

rank_summaries, 3, 24, 30

rPLMIX, 3, 31

selectPLMIX, 3, 32

unit_to_freq, 3, 35