Instructions

Here we show how to conduct rank aggregation using either MM, EMM or HEMM by examples. Users need to first install R package "ExtMallows". The R package includes four data examples and four functions: MM(), EMM(), HEMM() and corrRankings(). For any specific function, users can use help() to look for its details. For instance, help(EMM). Following are steps for doing rank aggregation using R package "ExtMallows".

(1) data preparation

The input data for MM(), EMM() and HEMM() should be a $n \times m$ matrix, with each column representing a ranking list that ranks the items from the most preferred to the least preferred. For the missing items, use 0 to denote them. Four data examples are given in the "data" folder.

(a) simu1: it is top-30 ranking, including 6 ranking lists. The data is generated as described in the simulation study 1 of the article.

(b) simu2: it is top-40 ranking, including 6 ranking lists. The data is generated as described in the simulation study 2 of the article.

(c) simu3: it includes 20 full rankings. The data is generated as described in the simulation study 3 of the article (highly correlated rankings, s=10).

(d) NBArankings: it includes 34 partial rankings. The data is a real NBA teams example that includes 6 full rankings from professional websites and 28 top-8 rankings collected from the surveys done by Harvard students. For details, please refer to the article.

(2) checking the independence of the rankings

To check the independence of the rankings, use corrRankings(). The input argument is the rankings, and the output is a symmetric matrix of p values that measure the correlation of pairwise rankings. The input rankings should have at least 8 ranking lists.

Example 1: check the independence of rankings

```
data(simu3)
pvalue=corrRankings(rankings = simu3)

#threshold the p values

threshold=0.05
pvalue.trunc=ifelse(pvalue<=0.05, pvalue, 1)

#plot the p values

x=y=1:ncol(pvalue)
par(mfrow=c(1,2))
image(x, y, pvalue, xlab = NA, ylab = NA, sub = "rank coefficient")
image(x, y, pvalue.trunc, xlab = NA, ylab = NA,
   sub = "rank coefficient < 0.05")
```

(3) If there are no over-correlated rankings and to apply the Mallows model or extended Mallows model, we use MM() and EMM() to do rank aggregation, respectively.

In MM(),

input arguments:

- rankings: a $n \times m$ matrix, with each column representing a ranking list.
- initial.method: the method for initializing $\boldsymbol{\pi}_0$, with four options: mean, median, geometric and random (the mean of three randomly sampled ranking lists). By default, initial.method = "mean".
- it.max: the maximum number of iterations. By default, it.max=20.

return arguments:

- op.phi: optimal value of $\boldsymbol{\phi}$
- op.pi0: optimal value of $\boldsymbol{\pi}_0$ that ranks the items from the most preferred to the least preferred.
- max.logL: maximum value of log-likelihood

Example 2: use MM() to aggregate rankings.

```
data(simu1)
res=MM(rankings = simu1, initial.method = "mean", it.max = 20)
res$op.phi
res$op.pi0
```

In EMM(),

input arguments:

- rankings: a $n \times m$ matrix, with each column representing a ranking list.
- initial.method: the method for initializing $\boldsymbol{\pi}_0$, with four options: mean, median, geometric and random (the mean of three randomly sampled ranking lists). By default, initial.method = "mean".
- it.max: the maximum number of iterations. By default, it.max=20.

return arguments:

- op.phi: optimal value of $\boldsymbol{\phi}$
- op.omega: optimal value of $\boldsymbol{\omega}$
- op.alpha: optimal value of $\boldsymbol{\alpha}$
- op.pi0: optimal value of $\boldsymbol{\pi}_0$ that ranks the items from the most preferred to the least preferred.
- max.logL: maximum value of log-likelihood

Example 3: use EMM() to aggregate independent rankings.

```
data(simu1)
res=EMM(rankings = simu1, initial.method = "mean", it.max = 20)
res$op.phi
res$op.omega
res$op.pi0
res$max.logL
```

(4) If there are over-correlated rankings, use HEMM() to do rank aggregation.

In HEMM(),

input arguments:

- rankings: a $n \times m$ matrix, with each column representing a ranking list.
- num.kappa: number of over-correlated ranking groups.
- is.kappa.ranker: a list of over-correlated ranking groups, with the k-th element denoting the column numbers of the rankings that belong to the k-th group.
- initial.method: the method for initializing the value of $\boldsymbol{\pi}_0$, with four options: mean, median, geometric and random (the mean of three randomly sampled ranking lists). By default, initial.method = "mean".
- it.max: the maximum number of iterations. By default, it.max=20.

return arguments:

- op.phi: optimal value of $\boldsymbol{\phi}$.
- op.phi1: optimal value of $\tilde{\boldsymbol{\phi}}$.
- op.omega: optimal value of $\boldsymbol{\omega}$
- op.alpha: optimal value of $\boldsymbol{\alpha}$
- op.pi0: optimal value of $\boldsymbol{\pi}_0$, ranking the items from the most preferred to the least preferred.

- op.kappa: optimal value of $\boldsymbol{\kappa}_1, ..., \boldsymbol{\kappa}_M$.
- max.logL: maximum value of log-likelihood.

Example 4: use HEMM to aggregate over-correlated rankings. For data simu3, there are two over-correlated ranking groups, the first one includes the first five rankings, and the second one includes the 6th-10th rankings.

```
data(simu3)
res=HEMM(rankings = simu3, num.kappa = 2, is.kappa.ranker = list(1:5, 6:10),
    it.max = 20)
res$op.phi
res$op.phi1
res$op.omega
res$op.pi0
```

Example 5: For data NBArankings, there are one over-correlated ranking groups, which includes the first six rankings.

```
data(NBArankings)
res=HEMM(rankings = NBArankings, num.kappa = 1, is.kappa.ranker = list(1:6),
    it.max = 20)
res$op.omega
res$op.pi0
res$op.kappa
```