Package ‘obliclus’

February 20, 2015

Type Package
Title Cluster-based factor rotation
Version 0.9
Date 2012-07-20
Author Michio Yamamoto
Maintainer Michio Yamamoto <michio.koko@gmail.com>
Description This package conducts factor rotation techniques which intend to identify a simple and well-clustered structure in a factor loading matrix.
License GPL (>= 2)
URL http://michioyamamoto.com
Repository CRAN
Date/Publication 2012-07-20 07:20:58
NeedsCompilation yes

R topics documented:

alpha.optim ................................................................. 2
analy.deriv ................................................................. 3
criteria ................................................................. 3
obliclus ................................................................. 4
random.cluster.C ..................................................... 6

Index 7
alpha.optim

Optimal value of alpha in Gradient Projection algorithm.

Description
This function gives a value of alpha which guarantees monotone decrease of the objective function in "obliclus".

Usage
alpha.optim(A, T, G, cluster, info, maxit = 1000)

Arguments
- **A**: The loading matrix for rotation.
- **T**: The current value of rotation matrix.
- **G**: The gradient of the objective function at T on the set of oblique rotation matrices.
- **cluster**: The vector of cluster parameters which indicate a cluster where each variable is assigned.
- **info**: The list including an initial value of alpha.
- **maxit**: The limit of the iteration of partial step modification for the value of alpha.

Value
The value of alpha which is calculated by the partial step modification described in Jennrich (2002).

Author(s)
Michio Yamamoto
<michio.koko@gmail.com>

References

See Also
obliclus
analy.deriv

---

**analy.deriv**

*Gradient of an objective function at a rotation matrix on the set of rotation matrices.*

---

**Description**

This function gives the gradient of an objective function at a rotation matrix $T$ on the set of rotation matrices.

**Usage**

`analy.deriv(A, T, cluster, info)`

**Arguments**

- **A**: The loading matrix for rotation.
- **T**: The current value of rotation matrix.
- **cluster**: which indicate a cluster where each variable is assigned.
- **info**: The list including `ncluster` the number of clusters, `nvar` and the value of an specific parameter to a rotation method, for example, the value of oblimin parameter for the method oblimin.

**Value**

The gradient matrix is returned.

**Author(s)**

Michio Yamamoto
<michio.koko@gmail.com>

**See Also**

`obliclus`

---

criteria

*The value of the objective function which is minimized.*

---

**Description**

The value of the objective function of Obliclus is calculated. In criteria, two different rotation criteria, Oblimin and Geomin, can be calculated in the objective function of Obliclus.

**Usage**

`criteria(A, T, cluster, info)`
Arguments

A  The loading matrix for rotation.
T  The current value of rotation matrix.
cluster  The vector of cluster parameters which indicate a cluster where each variable is assigned.
info  The list including N.cluster the number of clusters, N.var the number of observed variables, method a rotation method whose criterion is included in an objective function, and the value of an specific parameter to a rotation method, for example, the value of oblimin parameter for the method oblimin.

Value

The value of the objective function given A, T, and cluster.

Author(s)

Michio Yamamoto
<michio.koko@gmail.com>

References


See Also

obliclus

Description

This function conducts rotations of loading matrix based on cluster structure of variables. It can be used for two ways. One is to find simple and well-grouped structure simultaneously, and the other is to find simple structure based on the prior information on cluster structure. The objective function is optimized using the Gradient Projection (GP) algorithm and k-means algorithm alternately.

Usage

```r
obliclus(A, cluster = NULL, normalize = TRUE, N.random = 10,
    fixed = FALSE, N.cluster = NULL, ini.cluster = NULL,
    maxit = 1000, alpha = 1, method = "oblimin",
    geomin.par = 0.01, oblimin.index = 0, fit.cr = 1e-04)
```
**Arguments**

- **A**
  - The loading matrix for rotation.

- **cluster**
  - The vector of cluster parameters which indicate a cluster where each variable is assigned.

- **normalize**
  - If "TRUE", Kaiser's normalization is applied.

- **N.random**
  - The number of sets of random initial values of rotation matrix $T$.

- **fixed**
  - If "TRUE", obliclus gives an optimal loading matrix based on cluster structure cluster.

- **N.cluster**
  - The number of clusters. If this is null, the number of factors is used for the number of clusters.

- **ini.cluster**
  - The vector of initial cluster parameters.

- **maxit**
  - The upper limit of iteration of the algorithm.

- **alpha**
  - The initial value of alpha which is a step size in the GP algorithm.

- **method**
  - For "oblimin", obliclus conducts rotation whose objective function includes oblimin with oblimin.index and the measure of cluster similarity among variables. Similarly, for "geomin", obliclus conducts rotation whose objective function includes geomin with geomin.par and the measure of cluster similarity.

- **geomin.par**
  - The parameter for geomin rotation. The default is 0.01.

- **oblimin.index**
  - The parameter for oblimin rotation. The default is 0, which imply the Quartimin rotation.

- **fit.cr**
  - The criterion for convergence of algorithm, abs(criteria.new - criteria.old).

**Value**

- **A**
  - Rotated loading matrix.

- **T**
  - Estimated rotation matrix.

- **E**
  - Estimated correlation matrix among factors.

- **conv**
  - If "TRUE", the algorithm converges.

- **cluster**
  - Estimated cluster parameter.

- **cr**
  - The value of objective function finally attained.

**Author(s)**

Michio Yamamoto

<michio.koko@gmail.com>

**References**

random.cluster.C

Examples

### loading matrix for rotation (Perfect simple structure)
### find optimal T and U (cluster parameter) simultaneously
A <- matrix(c(.8, 0, 0,
                0, .8, 0,
                0, 0, .8,
                0, 0, .7), 6, 3, byrow=TRUE)

(out <- obliclus(A, N.random=1)) ##Using many random starts is recommended.

### loading matrix for rotation (More complex structure)
### Find optimal T based on the information on cluster structure
A <- matrix(c(1, 0, 0,
              0, 1, 0,
              0, 0, 1,
              0.9, 0.6,-0.3,
              -0.3, 0.9, 0.6,
              0.6,-0.3, 0.9) ,6,3,byrow=TRUE)
cluster <- c(1,2,3,4,4,4)
(out <- obliclus(A, cluster=cluster, fixed=TRUE, N.cluster=4))

random.cluster.C  Random cluster parameter

Description

This function generates a cluster parameter vector randomly. The upper limit of the number of clusters is 13.

Usage

random.cluster.C(info.to.C)

Arguments

info.to.C  The vector consisting of the number of variables, the number of clusters, a real number 1, and the minimum number of variables each cluster should have.

Value

The vector which includes a cluster number where each variable is assigned.

Author(s)

Michio Yamamoto
<michio.koko@gmail.com>
Index

alpha.optim, 2
analy.deriv, 3
criteria, 3
obliclus, 4
random.cluster.C, 6