

Package ‘OSDR’

January 2, 2019

Type Package

Title Finds an Optimal System of Distinct Representatives

Version 1.1.3

Author Massimo Cannas [aut, cre]

Maintainer Massimo Cannas <massimo.cannas@unica.it>

Description Provides routines for finding an Optimal System of Distinct Representatives (OSDR), as defined by D.Gale (1968) <doi:10.1016/S0021-9800(68)80039-0>.

Suggests optrees, optmatch

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2019-01-02 08:40:10 UTC

R topics documented:

| | |
|------------------------|---|
| OSDR-package | 1 |
| exdata | 3 |
| listmat | 4 |
| matlist | 5 |
| OSDR | 7 |

| | |
|--------------|-----------|
| Index | 11 |
|--------------|-----------|

| | |
|--------------|--|
| OSDR-package | <i>Finds an Optimal System of Distinct Representatives</i> |
|--------------|--|

Description

Provides routines for finding an Optimal System of Distinct Representatives (OSDR), as defined by D.Gale (1968) <doi:10.1016/S0021-9800(68)80039-0>.

Details

Package: OSDR
 Type: Package
 Title: Finds an Optimal System of Distinct Representatives
 Version: 1.1.3
 Authors@R: person("Massimo", "Cannas", role = c("aut", "cre"), email = "massimo.cannas@unica.it")
 Author: Massimo Cannas [aut, cre]
 Maintainer: Massimo Cannas <massimo.cannas@unica.it>
 Description: Provides routines for finding an Optimal System of Distinct Representatives (OSDR), as defined by D.Gale (19
 Suggests: optrees, optmatch
 License: GPL-2

Index of help topics:

| | |
|--------------|---|
| OSDR | Finds an Optimal System of Distinct Representatives from a family of subsets. |
| OSDR-package | Finds an Optimal System of Distinct Representatives |
| exdata | Executive dataset (subset of) |
| listmat | From cost matrix to list of feasible assignments. |
| matlist | From list of feasible assignments to cost matrix. |

The aim of this small package is to provide a function solving the assignment problem on an *ordered* set. This matching problem can be viewed in two ways. First, as the original ordered assignment problem, where a set of jobs, ordered by importance, must be filled by suitable applicants in the best possible way. Second, as an ordered matching problem where treated unit should be matched in order of importance with suitable controls. Suitability can be obtained by a "caliper": units within the caliper having zero matching cost and units outside the caliper having infinity cost. The main function OSDR exploits an algorithm suggested by I.Anderson to find an *order optimal* matching, as defined by D.Gale (see [OSDR](#) for details). The package includes some utilities and examples (both combinatorial and statistically oriented) to illustrate the use of OSDR.

Author(s)

Massimo Cannas [aut, cre]

Maintainer: Massimo Cannas <massimo.cannas@unica.it>

References

Gale, D. (1968) Optimal matching in an ordered set: an application of matroid theory. *Journal of Combinatorial Theory* 4, pp. 176-180.

Anderson, I. (1989) A first course in Combinatorial Mathematics. *Oxford University Press*.

Rosenbaum, P. R. (1989). Optimal matching for observational studies. *Journal of the American Statistical Association* 84(408): pp. 1024-1032.

Cannas, M. Order optimal matching for statistical application: a gender gap case study (soon available on *arXiv*).

See Also

Routines for classic combinatorial optimization problems are available on R via the [optrees](#) package and the [maxmatching](#) package. A package for statistically oriented optimal matching is the [optmatch](#) package which can be used in observational studies to find a minimum covariate distance matching of control to treated units prior to outcome analysis (Rosenbaum).

Examples

```
#See OSDR help for the examples.
```

| | |
|--------|--------------------------------------|
| exdata | <i>Executive dataset (subset of)</i> |
|--------|--------------------------------------|

Description

Contains executives data for an italian firm operating in the energy sector (year 2015). The complete dataset contains 302 rows (10 firms) and 180 columns (it will be released in next package version).

Usage

```
data("exdata")
```

Format

A data frame with 28 observations on the following 8 variables.

firm a character vector
position a numeric vector
education a numeric vector
year_born a numeric vector
contract a numeric vector
part_fulltime a numeric vector
seniority a numeric vector
sex a numeric vector

Details

position is coded: 4 = top manager, 3 = medium/first line manager, 2 = supervisor

sex is coded: 0 = M, 1 = F

education is coded: Post-graduate = 5, Graduate = 4, High school = 3

contract type is coded: fixed term = 4; permanent = 3

seniority is measured by number of years in the current position

References

Cannas, M. Order optimal matching for statistical application: a gender gap case study (soon available on *arXiv*).

Examples

```
# load executive data
data(exdata)

# case study: matched samples for comparing women and men executives

table(exdata$sex)
table(exdata$position,exdata$sex)
# There are more women and more in apical position.

# A complete matching is not possible for several choices of the caliper.
# Gap differences tend to be higher for higher ranks

# e.g. Lynn and Thompson(1997), Above the glass ceiling? A comparison of
# matched Samples of Men and Women Executives. J. of Appl. Psych. 82(3)

# so we would give higher matching priority to women in higher position.
# We can use OSDR to find a minimum cost matching
# performing matching by decreasing hierarchical position.

# An illustrative example is shown in the help page of the OSDR function.
```

listmat

From cost matrix to list of feasible assignments.

Description

Transforms the cost matrix of the assignment problem in the corresponding list of suitable applicants for each job.

Usage

```
listmat(x)
```

Arguments

x A cost matrix where entry (i, j) is the cost of assigning job i to applicant j .

Details

In statistical matching problems the input is usually the cost matrix while in assignment problems is the list of assignable elements. Functions `matlist` and `listmat` go back and forth these two representations of the input.

Value

A list of suitable applicants. The i^{th} element of the list contains the suitable applicants for job i . These are all applicants with *finite* assignment cost. Note that if the cost matrix contains finite non zero values there is a loss of information in the transformation.

Author(s)

Massimo Cannas <massimo.cannas@unica.it>

See Also

See [matlist](#), the reverse function.

Examples

```
# a list of feasible applicants for five jobs
M1<-c("A","B","C")
M2<-c("A","C")
M3<-c("B")
M4<-c("A","C")
M5<-c("A","D")
M <-list(M1,M2,M3,M4,M5)
M

# list --> cost matrix
m <- listmat(M)

# cost matrix --> list
l <- matlist(m)
```

`matlist`

From list of feasible assignments to cost matrix.

Description

Transforms the list of suitable applicants into the equivalent cost matrix.

Usage

```
matlist(x)
```

Arguments

`x` An ordered list of applicants for a set of jobs. The i^{th} element of the list contains applicants suitable for job i . Equivalently, an ordered list of controls where the i^{th} element of the list contains controls matchable with the i^{th} treated unit.

Details

In statistical matching problems the input is usually the cost matrix while in assignment problems is the list of assignable elements. Functions `matlist` and `listmat` go back and forth these two representations of the input.

Value

The cost matrix of the assignment problem. The (i, j) entry of the matrix is zero if the j^{th} applicant is assignable to the i^{th} job and ∞ otherwise. Equivalently, the (i, j) entry of the matrix is zero if the j^{th} control unit can be matched to the i^{th} treated unit and ∞ otherwise.

Author(s)

Massimo Cannas <massimo.cannas@unica.it>

See Also

See also the reverse function [listmat](#).

Examples

```
# a list of feasible applicants for five jobs
M1<-c("A","B","C")
M2<-c("A","C")
M3<-c("B")
M4<-c("A","C")
M5<-c("A","D")
M <-list(M1,M2,M3,M4,M5)
M

# the corresponding cost matrix
m<-listmat(M)

# back to the list
l<-matlist(m)
```

| | |
|------|--|
| OSDR | <i>Finds an Optimal System of Distinct Representatives from a family of subsets.</i> |
|------|--|

Description

The function finds an Optimal Set of Distinct Representatives (OSDR) from an ordered family of subsets. Optimality is order-wise in the sense specified by Gale: any other SDR cannot be larger and its elements cannot be chosen from more important sets (see details).

Usage

OSDR(M)

Arguments

M An ordered family of subsets. In assignment problems the list specifying the feasible applicants for each job. In matching problems the list specifying the controls matchable with each treated unit. The list is assumed *ordered* by decreasing assignment/matching priority.

Details

Consider a set of jobs and a set of suitable applicants for each job. It was shown by D.Gale that there exists an *optimally assignable* subset of the jobs in the sense that if a_1, \dots, a_k is optimal and b_1, \dots, b_h is another assignable set of jobs then: a) $k \geq h$ and b) $a_i \leq b_i$ for all i . The latter means that job a_i has as higher a priority as job b_i . From a graph perspective, if J is the set of jobs and A is the set of applicants the OSDR is an *order wise* maximum size matching of the bipartite graph with vertex set $Union A$.

Function OSDR finds the optimal matching using an algorithmic proof of Hall's theorem due to logician D.J. Shoesmith. The algorithm assigns greedily until possible and correct backward when necessary. A message is given when a correction is attempted (when it fails another message warns that Hall's condition is not satisfied).

Value

A list containing three elements: the OSDR (a zero in position i indicates impossibility of assigning job i), the index of optimally assignable jobs and the index of unassignable jobs. If the latter is not empty M does not meet Hall's condition, i.e., a complete SDR is not possible.

Note

In statistical matching problems the sets J and A are the sets of treated and control units and it is usually required to find the minimum covariate distance matching of treated versus control units. When a complete matching does not exist it can be convenient to find either an order optimal matching or a minimum cost matching on the OSDR subset (see the gender gap example).

Author(s)

Massimo Cannas <massimo.cannas@unica.it>

References

Gale, D. (1968) Optimal matching in an ordered set: an application of matroid theory. *Journal of Combinatorial Theory 4: 176-180*.

Anderson, I. (1989) A first course in Combinatorial Mathematics. *Oxford University Press*.

See Also

See also [matlist](#), [listmat](#)

Examples

```
### example 1
# M is the list of suitable applicants for five jobs
M1 <- c("A" , "B")
M2 <- c("B" , "C")
M3 <- c("B")
M4 <- c("A" , "C")
M5 <- c("B" , "C" , "D")
M <- list(M1 , M2 , M3 , M4 , M5)

OSDR(M)

# $OSDR
# [1] "A" "C" "B" "0" "D"
# $matched
# [1] 1 2 3 5
# $unmatched
# [1] 4

# job 4 unmatched so Hall's condition is not satisfied: it's impossible to fill all the jobs
# note that there are (order-\emph{suboptimal}) assignments of the same length of the optimal:
# eg: 0CBAD , BC0AD

#### example 2: slightly modified: more than one order optimal matching
M1<-c("A","B","C")
M2<-c("A","C")
M3<-c("B")
M4<-c("A","C")
M5<-c("A","D")
M <-list(M1,M2,M3,M4,M5)

OSDR(M)

# note there are other order optimal matchings: ACB0D or CAB0D
# note there are also other maximum size matchings (not order optimal):
# e.g. 0CBAD or BC0AD
```



```

#### Case Study: matching men and women executives
# load executive data
data(exdata)

# descriptives on:
# sex(0=M; 1=F) ;
# position (4=top manager, 3=medium/first line manager, 2 =supervisor);
table(exdata$sex)# there are more women
table(exdata$position,exdata$sex)# and more in apical position

# order by matching priority (high-rank women first)
# see e.g. Lynn and Thompson(1997), J. of Appl. Psych. 82(3)
data <- exdata[order(-exdata$sex,-exdata$position, -exdata$seniority),]

# covariate distance matrix
require(optmatch)
dist <- match_on(sex ~ position+education+year_born+contract+part_fulltime+seniority ,
data=exdata)
# use broad caliper to avoid very bad matches
dist <- caliper(dist,4,values=TRUE)

# minimum distance pair matching (package optmatch)
copt <- pairmatch(dist,data=exdata)
summary(copt)
sum(matched.distances(copt,dist)) # total cost 19

### find osdr
#order dist by priority order (i.e by decreasing position)
dist <- as.matrix(dist)[order(match(rownames(dist),rownames(exdata))),]
mylist <- matlist(dist)
res <- OSDR(mylist)

# index and labels of treated and untreated in OSDR
ord_dist<-as.matrix(dist)[order(match(names(mylist),rownames(exdata))),]
index_t<-res$matching; names_t<-rownames(ord_dist)[index_t]
index_ut<-res$unmatched;names_ut<-rownames(ord_dist)[index_ut]

# compare matched treated: optmatch vs ordmatch
#matched treated optmatch
  matched(copt)[which(exdata$sex==1)]
#matched treated ordmatch
  rownames(data)[which(exdata$sex==1)] %in% names_t

# compare total matching cost: optmatch vs ordmatch

# case 1: distance matrix is zero infinity: same cost (0)
# case 2: distance matrix is not zero infinity
#       find minimum cost matching on osdr:

data2<-exdata[-match(names_ut,rownames(exdata)),]
dist2<-as.matrix(dist)[-match(names_ut,rownames(dist)),]
copt2 <- pairmatch(dist2,data=data2)

```

```
summary(copt2);copt2  
sum(matched.distances(copt2,dist2)) # 22  
sum(matched.distances(copt,dist)) # 19
```

Index

- *Topic **\textasciitilde assignment cost matrix**
 - matlist, 5
 - *Topic **\textasciitilde assignment problem**
 - matlist, 5
 - *Topic **\textasciitildeassignment problem**
 - listmat, 4
 - *Topic **\textasciitildeincomplete matching**
 - OSDR, 7
 - *Topic **\textasciitildelist of feasible assignments**
 - listmat, 4
 - *Topic **\textasciitildeorder optimal matching**
 - OSDR, 7
 - *Topic **datasets, matched pair comparison, gender gap, observational studies**
 - exdata, 3
 - *Topic **ordered assignment problem, order optimal matching, observational studies.**
 - OSDR-package, 1
- exdata, 3
- listmat, 4, 6, 8
- matlist, 5, 5, 8
- maxmatching, 3
- optmatch, 3
- optrees, 3
- OSDR, 2, 7
- OSDR-package, 1