Package ‘OutrankingTools’

February 19, 2015

Type Package

Title Functions for Solving Multiple-criteria Decision-making Problems

Version 1.0

Date 2014-12-22

Author Michel Prombo <michel.prombo@statec.etat.lu> or <michel.prombo@gmail.com>

Maintainer Michel Prombo <michel.prombo@graduates.centraliens.net>

Description Functions to process "outranking" ELECTRE methods existing in the literature. See, e.g., <http://en.wikipedia.org/wiki/ELECTRE> about the outranking approach and the foundations of ELECTRE methods.

Depends igraph

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2014-12-24 01:26:42

R topics documented:

OutrankingTools-package ................................................. 2
Electre3_AlphaBetaThresholds .......................................... 2
Electre3_SimpleThresholds ............................................. 5
Electre_1 ................................................................. 7
Electre_tri ............................................................... 9

Index 13
OutrankingTools-package

Functions for Solving Multiple-criteria Decision-making Problems

Description

The outranking methods constitute one of the most fruitful approach in the field of Multiple Criteria Decision Making (MCDM). They main feature is to compare all feasible alternatives or actions by pair building up some binary relations, crisp or fuzzy, and then exploit in appropriate way these relations in order to obtain final recommendations. This package contains functions to process ELECTRE methods existing in the literature. See, e.g., <http://en.wikipedia.org/wiki/ELECTRE> about the outranking approach and the foundations of ELECTRE methods.

Details

Package: OutrankingTools
Type: Package
Version: 1.0
Date: 2014-12-22
License: GPL (>= 2)

Author(s)

Michel Prombo <michel.prombo@statec.etat.lu>
Maintainer: <michel.prombo@graduates.centraliens.net>

References


Electre3_AlphaBetaThresholds

ELECTRE III using affine function form of the thresholds
Description

ELECTRE III method aims to answer the following question: considering a finite set of actions, A, evaluated on a coherent family of pseudo-criteria, F, how to make a partition of A in classes of equivalence and provide a necessarily complete pre-order expressing the relative position of these classes? In the first phase, ELECTRE III method involves the construction of a fuzzy outranking relation. In the second phase, an algorithm is used for making a ranking in a final partial pre-order, that combines two complete pre-orders.

Usage

Electre3_AlphaBetaThresholds(performanceMatrix,
alternatives,
criteria,
minmaxcriteria,
criteriaWeights,
alpha_q,
beta_q,
alpha_p,
beta_p,
alpha_v,
beta_v,
mode_def)

Arguments

performanceMatrix Matrix or data frame containing the performance table. Each row corresponds to an alternative, and each column to a criterion. Rows (resp. columns) must be named according to the IDs of the alternatives (resp. criteria).

alternatives Vector containing names of alternatives, according to which the data should be filtered.

criteria Vector containing names of criteria, according to which the data should be filtered.

minmaxcriteria criteriaMinMax Vector containing the preference direction on each of the criteria. "min" (resp."max") indicates that the criterion has to be minimised (maximised).

criteriaWeights Vector containing the weights of the criteria.

alpha_q Vector containing the coefficients alpha when indifference threshold is as affine function of the performance.

beta_q Vector containing coefficients beta when indifference threshold is as affine function of the performance.

alpha_p Vector containing coefficients beta when preference threshold is as affine function of the performance.

beta_p Vector containing coefficients beta when preference threshold is as affine function of the performance.
alpha_v  Vector containing coefficients beta when veto threshold is as affine function of the performance.

beta_v  Vector containing coefficients beta when veto threshold is as affine function of the performance.

mode_def  Vector containing the mode of definition which indicates the mode of calculation of the thresholds (direct (D), considers the worst of the two actions; inverse(I), considers the best of the two actions). If Null, "Direct" mode will be setting

Author(s)
Michel Prombo <michel.prombo@statec.etat.lu>

References

Examples

## Illustrative example used to present the ELECTRE III-IV software in the French version.
## The objective: make the ranking of 10 French cars that were evaluated on 7 criteria
## Document du LAMSADE 85, Universite Paris-Dauphine,Paris.)

## the performance table

performanceMatrix <- cbind(c(103000,101300,156400,267400,49900,103600,103000,170100,279700,405000),
                          c(171.3,205.3,221.7,230.7,122.6,205.1,178.0,226.0,233.8,265.0),
                          c(7.65,7.90,7.90,10.50,8.30,8.20,7.20,9.10,10.90,10.30),
                          c(352,203,391,149,120,265,419,419,359,265),
                          c(11.6,8.4,4.8,6.23,7.8.1,11.4,8.1,7.8,6.0),
                          c(88.0,78.3,81.5,64.7,74.1,81.7,77.6,74.7,75.5,74.7),
                          c(69.7,73.4,69.0,65.6,76.4,73.6,66.2,71.7,70.9,72.0))

# Vector containing names of alternatives
alternatives<-c("CBX16","P205G","P405M","P605S","R4GTL","RCLIO","R21TS","R21TU","R25BA","ALPIN")

# Vector containing names of criteria
criteria <-c("Prix","Vmax","Cl20","Coff","Acce","Frei","Brui")
# vector indicating the direction of the criteria evaluation.
minmaxcriteria <-c("min","max","min","min","max","min","min")

# criteriaWeights vector
criteriaWeights <- c(0.3,0.1,0.3,0.2,0.1,0.2,0.1)

# thresholds vector
alpha_q <- c(0.08,0.02,0,0,0.1,0,0)
Electre3_SimpleThresholds

beta_q <- c(-2000, 0, 1, 100, -0.5, 0, 3)
alpha_p <- c(0.13, 0.05, 0, 0, 0.2, 0, 0)
beta_p <- c(-3000, 0, 2, 200, -1.5, 5)
alpha_v <- c(0.9, NA, 0, NA, 0.5, 0, 0)
beta_v <- c(50000, NA, 4, NA, 3, 15, 15)

# Vector containing the mode of definition which
# indicates the mode of calculation of the thresholds.
mode_def <- c("I", "D", "D", "D", "D", "D", "D")

# Testing
Electre3_AlphaBetaThresholds(performanceMatrix, alternatives, criteria, minmaxcriteria, criteriaWeights, alpha_q, beta_q, alpha_p, beta_p, alpha_v, beta_v, mode_def)

Electre3_SimpleThresholds

**ELECTRE III using non affine form of the thresholds**

**Description**

ELECTRE III method aims to answer the following question: considering a finite set of actions, A, evaluated on a coherent family of pseudo-criteria, F, how to make a partition of A in classes of equivalence and provide a necessarily complete pre-order expressing the relative position of these classes? In the first phase, ELECTRE III method involves the construction of a fuzzy outranking relation. In the second phase, an algorithm is used for making a ranking in a final partial pre-order, that combines two complete pre-orders.

**Usage**

Electre3_SimpleThresholds(performanceMatrix, alternatives, criteria, minmaxcriteria, criteriaWeights, IndifferenceThresholds, PreferenceThresholds, VetoThresholds, mode_def)
Arguments

performanceMatrix
Matrix or data frame containing the performance table. Each row corresponds to an alternative, and each column to a criterion. Rows (resp. columns) must be named according to the IDs of the alternatives (resp. criteria).

alternatives
Vector containing names of alternatives, according to which the data should be filtered.

criteria
Vector containing names of criteria, according to which the data should be filtered.

minmaxcriteria
criteriaMinMax Vector containing the preference direction on each of the criteria. "min" (resp."max") indicates that the criterion has to be minimized (maximized).

criteriaWeights
Vector containing the weights of the criteria.

IndifferenceThresholds
Vector containing the indifference thresholds constraints defined for each criterion.

PreferenceThresholds
Vector containing the preference thresholds constraints defined for each criterion.

VetoThresholds
Vector containing the veto thresholds constraints defined for each criterion

mode_def
Vector containing the mode of definition which indicates the mode of calculation of the thresholds (direct (D), considers the worst of the two actions; inverse(I), considers the best of the two actions). If Null, "Direct" mode will be setting

Author(s)
Michel Prombo <michel.prombo@statec.etat.lu>

References

Examples

# the performance table
performanceMatrix <- cbind(  
  c(-14,129,-10,44,-14),  
  c(90,100,50,90,100),  
  c(0,0,0,0,0),  
  c(40,0,10,5,20),  
  c(100,0,100,20,40)  
)

# Vector containing names of alternatives
Electre_1

alternatives <- c("Project1","Project2","Project3","Project4","Project5")

# Vector containing names of criteria
criteria <- c("CR1","CR2","CR3","CR4","CR5")

# vector indicating the direction of the criteria evaluation
minmaxcriteria <- c("max","max","max","max","max")

# criteriaweights vector
# thresholds vector
IndifferenceThresholds <- c(25,16,0,12,10)
PreferenceThresholds <- c(50,24,1,24,20)
VetoThresholds <- c(100,60,2,48,90)
criteriaweights <- c(1,1,1,1,1)

# Vector containing the mode of definition which
# indicates the mode of calculation of the thresholds.

# Testing
Electre1_SimpleThresholds(performanceMatrix, alternatives, criteria, minmaxcriteria, criteriaweights, IndifferenceThresholds, PreferenceThresholds, VetoThresholds)

---

Electre_1

**Electre 1 : Method used to solve multiple criteria decision making**

Description

The acronym ELECTRE stands for: ELimination Et Choix Traduisant la RÉalitë (ELimination and Choice Expressing REality). ELECTRE I method is then designed to rank reliability design scheme in order of decision maker preference. This method is based on the concept of concordance and discordance.

Usage

Electre_1(performanceMatrix, alternatives, criteria, criteriaweights,
minmaxcriteria,
concordance_threshold = 1,
discordance_threshold = 0)

Arguments

performanceMatrix
Matrix or data frame containing the performance table. Each row corresponds
to an alternative, and each column to a criterion. Rows (resp. columns) must be
named according to the IDs of the alternatives (resp. criteria).

alternatives
Vector containing names of alternatives, according to which the data should be
filtered.

criteria
Vector containing names of criteria, according to which the data should be fil-
tered.

criteriaweights
tor containing the weights of the criteria.

minmaxcriteria
criteriaMinMax Vector containing the preference direction on each of the crite-
ria. "min" (resp."max") indicates that the criterion has to be minimized (maxi-
mized).

concordance_threshold
parameter defining concordance threshold . The default value is 1. The user can
set a new value between 0 and 1

discordance_threshold
parameter defining discordance threshold . The default value is 0. The user can
set a new value between 0 and 1.

Value

The function returns a list structured as follows :

"Performance Matrix"
A matrix containing the performance table. Each row corresponds to an alterna-
tive, and each column to a criterion

"Concordance Matrix"
Concordance matrix is one of two working relations (concordance and dis-
cordance) which are subsequently used to construct the final dominance rela-
tion. For an outranking aSb to be validated, a sufficient majority of criteria should
be in favor of this assertion.

"Discordance Matrix"
Discordance matrix is one of two working relations (concordance and discor-
dance) which are subsequently used to construct the final dominance relation.
The concept of discordance is complementary to the one of (concordance and
represents the discomfort experienced in the choosing of alternative a above al-
ternative b

Author(s)

Michel Prombo <michel.prombo@statec.etat.lu>
Electre_tri

References


Examples

```r
## This illustrative example has been used in to present
## the ELECTRE III-IV software in the French version.
## The objective is to make the ranking of 10 French cars that were evaluated on 7 criteria
## Document du LAMSADE 85, Universite Paris-Dauphine,Paris.)

## The performance table

performanceMatrix <- cbind(
  c(103000, 101300, 156400, 267400, 49900, 103600, 103000, 170100, 279700, 405000),
  c(171.3, 205.3, 221.7, 230.7, 122.6, 205.1, 178.0, 226.0, 233.8, 265.0),
  c(7.65, 7.90, 7.90, 10.50, 8.30, 8.20, 7.20, 9.10, 10.90, 10.30),
  c(352, 203, 391, 149, 120, 265, 419, 419, 395, 265),
  c(11.6, 8.4, 8.4, 8.6, 23.7, 8.1, 11.4, 8.1, 7.8, 6.0),
  c(88.0, 78.3, 81.5, 64.7, 74.1, 81.7, 77.6, 74.7, 75.5, 74.7),
  c(69.7, 73.4, 69.0, 65.6, 76.4, 73.6, 66.2, 71.7, 70.9, 72.0)
)

## Vector containing names of alternatives


## Vector containing names of criteria

criteria <- c("Prix","Vmax","C120","Coff","Acce","Frei","Brui")

## vector indicating the direction of the criteria evaluation

minmaxcriteria <- c("min","max","min","max","min","min","min")

## criteriaweights vector

 criteriaweights <- c(0.3,0.1,0.3,0.2,0.1,0.2,0.1)

Electre_tri(performanceMatrix, alternatives, criteria, criteriaweights, minmaxcriteria,
concordance_threshold=0.8, discordance_threshold=0.1)
```

Electre_tri

ELECTRE TRI Method

Description

The Electre Tri is a multiple criteria decision aiding method, designed to deal with sorting problems. Electre Tri method has been developed by LAMSADE (Paris-Dauphine University, Paris, France).
Usage

```
Electre_tri(performanceMatrix, alternatives, profiles, profiles_names, criteria, minmaxcriteria, criteriaWeights, IndifferenceThresholds, PreferenceThresholds, VetoThresholds, lambda = NULL)
```

Arguments

- **performanceMatrix**: Matrix or data frame containing the performance table. Each row corresponds to an alternative, and each column to a criterion. Rows (resp. columns) must be named according to the IDs of the alternatives (resp. criteria).
- **alternatives**: Vector containing names of alternatives, according to which the data should be filtered.
- **profiles**: Matrix containing, in each row, the lower profiles of the categories. The columns are named according to the criteria, and the rows are named according to the categories. The index of the row in the matrix corresponds to the rank of the category.
- **profiles_names**: Vector containing profiles' names.
- **criteria**: Vector containing names of criteria, according to which the data should be filtered.
- **minmaxcriteria**: criteriaMinMax Vector containing the preference direction on each of the criteria. "min" (resp."max") indicates that the criterion has to be minimized (maximized).
- **criteriaweights**: Vector containing the weights of the criteria.
- **indifferencethresholds**: Vector containing the indifference thresholds constraints defined for each criterion.
- **preferencethresholds**: Vector containing the preference thresholds constraints defined for each criterion.
- **vetothresholds**: Vector containing the veto thresholds constraints defined for each criterion.
- **lambda**: The lambda-cutting lambda- should be in the range 0.5 and 1.0) level indicates how many of the criteria have to be fulfilled in order to assign an alternative to a specific category. Default value=0.75

Author(s)

Michel Prombo <michel.prombo@statec.etat.lu>
References

Examples

# the performance table

performanceMatrix <- cbind(
c(-120.0,-150.0,-100.0,-60,-30.0,-80,-45.0),
c(-284.0,-269.0,-413.0,-596,-1321.0,-734,-982.0),
c(5.0,2.0,4.0,6.0,8.0,5.7,0),
c(3.5,4.5,5.5,8.7,5.4,8.5),
c(18.0,24.0,17.0,20,16.0,21,13.0)
)
# Vector containing names of alternatives
alternatives <- c("a1","a2","a3","a4","a5","a6","a7")

# Vector containing names of criteria

criteria <- c("g1","g2","g3","g4","g5")
criteriaWeights <- c(0.25,0.45,0.10,0.12,0.08)

# vector indicating the direction of the criteria evaluation
minmaxcriteria <- c("max","max","max","max","max")

# Matrix containing the profiles
profiles <- cbind(c(-100,-50),c(-1000,-500),c(4,7),c(4,7),c(15,20))
# vector defining profiles' names
profiles_names <-c("b1","b2")

# thresholds vector
IndifferenceThresholds <- c(15,80,1,0.5,1)
PreferenceThresholds <- c(40,350,3,3.5,5)
VetoThresholds <- c(100,850,5,4.5,8)

# Testing
Electre_tri(performanceMatrix,
alternatives,
profiles,
profiles_names,
criteria,  
minmaxcriteria,  
criteriaWeights,  
IndifferenceThresholds,  
PreferenceThresholds,  
VetoThresholds,  
lambda=NULL)

```r

```
Index

*Topic **Aggregation/disaggregation approaches**
  Electre_tri, 9

*Topic **Discrimination thresholds**
  Electre3_AlphaBetaThresholds, 2
  Electre3_SimpleThresholds, 5

*Topic **ELECTRE methods**
  Electre3_AlphaBetaThresholds, 2
  Electre3_SimpleThresholds, 5
  Electre_tri, 9
  OutrankingTools-package, 2

*Topic **Multi-criteria decision aiding**
  Electre_tri, 9

*Topic **Outranking approaches**
  Electre3_AlphaBetaThresholds, 2
  Electre3_SimpleThresholds, 5
  OutrankingTools-package, 2

*Topic **Sorting problem**
  Electre_tri, 9
  OutrankingTools-package, 2

*Topic `textasciitildekekwd1`
  Electre_1, 7

*Topic `textasciitildekekwd2`
  Electre_1, 7

*Topic **package**
  OutrankingTools-package, 2

*Topic **preference modelling, multicriteria analysis**
  Electre3_AlphaBetaThresholds, 2
  Electre3_SimpleThresholds, 5
  OutrankingTools-package, 2

*Topic **pseudo-criterion**
  Electre3_AlphaBetaThresholds, 2
  Electre3_SimpleThresholds, 5

Electre3_AlphaBetaThresholds, 2
Electre3_SimpleThresholds, 5
Electre_1, 7
Electre_tri, 9