

# Algorithmic Decision Theory

## Lecture 3: On consensual social ranking

Raymond Bisdorff

FSTC - CSC/ILIAS

4 mars 2020

### Content

1. On ranking from different opinions
  - Definition of the ranking problem
  - Linear Rankings
  - Majority margins
2. Types of ranking rules
  - Borda type rules
  - Condorcet : Ranking-by-choosing rules
  - Condorcet : Ranking-by-scoring rules
3. A classification of ranking rules
  - Condorcet-consistency
  - M-ordinality and M-invariance
  - Which ranking rule should we use?

### Definition of the ranking problem

A ranking rule is a procedure which aggregates marginal, ie individual voters, experts or criteria based, rankings into a global *consensus ranking* which combines the available preferential information *best* from the marginal viewpoints.

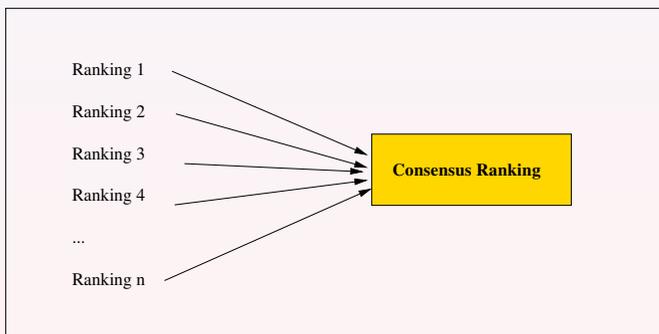


FIGURE – 1. Computing a consensual ranking

### Example (A first example : Borda's average ranks)

```

>>> from votingProfiles import *
>>> v = LinearVotingProfile('example1')
>>> v.showLinearBallots()
  voters      marginal candidate's
(weight)      rankings
v1(8):  ['a', 'c', 'b', 'e', 'd']
v2(7):  ['e', 'b', 'c', 'd', 'a']
v3(4):  ['d', 'c', 'b', 'e', 'a']
v4(4):  ['b', 'd', 'e', 'c', 'a']
v5(2):  ['c', 'd', 'b', 'e', 'a']
# voters: 25
>>> v.showRankAnalysisTable()
-----
candi- | Borda rank analysis tableau -----
dates | candidate x rank | Borda
-----|-----|-----
      | 1 2 3 4 5 | score average
-----|-----|-----
'b'   | 4 7 14 0 0 | 60 2.40
'c'   | 2 12 7 4 0 | 63 2.52
'e'   | 7 0 4 14 0 | 75 3.00
'd'   | 4 6 0 7 8 | 84 3.36
'a'   | 8 0 0 0 17 | 93 3.72
  
```

## Linear Rankings

- A **linear ranking**  $R = [a_1, a_2, \dots, a_n]$  is a list of  $n$  objects (a set  $X$  of candidates or decision alternatives) where the indexes  $1 \leq i < j \leq n$  represent a complete preferential ' $a_i$  **better than**  $a_j$ ' relation without ties ( $a_i > a_j$ ). The reversed list is called a **linear order**.
- A linear ranking  $R$  may be modelled with the help of a **bipolar characteristic function**  $r(a_i > a_j) \in \{-1, 0, 1\}$  where :

$$r(a_i > a_j) = \begin{cases} +1 & \text{if } i < j, \\ -1 & \text{if } i > j, \\ 0 & \text{otherwise.} \end{cases}$$

- Notice that reversing a ranking  $R$  is achieved by negation :  $r(a_i \not> a_j) = -r(a_i > a_j)$  which characterizes the corresponding linear order.

## Properties of linear rankings

A **linear ranking**  $R = [a_1, a_2, \dots, a_n]$  is

- a **transitive** relation,  $\forall i, j, k = 1..n$  :  
 $[(r(a_i > a_j) = +1) \wedge (r(a_j > a_k) = +1)] \Rightarrow (r(a_i > a_k) = +1)$ ;
- a **complete** relation,  $\forall i \neq j$  :  
 $r((a_i > a_j) \vee (a_j > a_i)) = \max(r(x_i > x_j), r(x_j > x_i)) = +1$ ;
- an **irreflexive** relation,  $\forall i$  :  
 $r(a_i > a_i) = 0$  /\* We ignore the reflexive relations \*/.
- A ranking with ties –a collection of ordered equivalence classes– is called a **weak ranking**; its **converse** is called a **preorder**, and its **negation** is called a **weak order**.

## Majority margins

The **majority margin**  $M(x, y)$  counts the *net advantage* of a candidate  $x$  over a candidate  $y$ . With  $k$  voters :

$$\begin{aligned} M(x, y) &= \sum_{k=1}^n (r(x >_k y)) + \sum_{k=1}^n (r(y \not>_k x)) \\ &= \sum_{k=1}^n [r(x >_k y) - r(y >_k x)] \end{aligned}$$

If the profile  $u$  consist of **complete** linear rankings, then :

$$M(x, x) = 0 \quad \text{and} \quad M(x, y) + M(y, x) = 0.$$

In this case, indeed :

$$\sum_{k=1}^n (r(x >_k y)) = n - \sum_{k=1}^n (r(y >_k x))$$

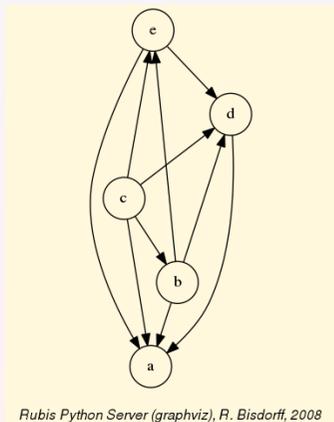
## Example (Computing majority margins)

```
>>> v
Instance class   : LinearVotingProfile
Instance name    : example1
# Candidates: 5, # Voters: 5
>>> v.showLinearBallots()
coalition      marginal candidate's
(weight)        rankings
v1(8):         ['a', 'c', 'b', 'e', 'd']
v2(7):         ['e', 'b', 'c', 'd', 'a']
v3(4):         ['d', 'c', 'b', 'e', 'a']
v4(4):         ['b', 'd', 'e', 'c', 'a']
v5(2):         ['c', 'd', 'b', 'e', 'a']
Total number of voters: 25
>>> from votingProfiles import CondorcetDigraph
>>> cd = CondorcetDigraph(v)
>>> cd.showMajorityMargins()
* ---- Relation Table ----
M(x,y) | 'a' 'b' 'c' 'd' 'e'
-----|-----
'a'    | 0  -9  -9  -9  -9
'b'    | 9   0  -3  13  11
'c'    | 9   3   0   9   3
'd'    | 9  -13  -9   0  -5
'e'    | 9  -11  -3   5   0
Valuation domain: [-25;+25]
```

The **majority relation**  $C(x, y)$  checks if a majority margin  $M(x, y)$  is *positive*, ie if there is a majority of rankings which rank candidate  $x$  before candidate  $y$  :

$$C(x, y) = \begin{cases} +1 & \text{if } M(x, y) > 0 \\ -1 & \text{if } M(x, y) < 0 \\ 0 & \text{otherwise.} \end{cases}$$

```
>>> cdp = PolarisedDigraph(cd, level=0, \
    StrictCut=True, KeepValues=False)
>>> cdp.recodeValuation(-1, 1)
>>> cdp.showRelationTable(ndigits=0)
* ---- Relation Table ----
C(x,y) | 'a'  'b'  'c'  'd'  'e'
-----|-----
'a'    |  0   -1  -1  -1  -1
'b'    |  1    0  -1   1   1
'c'    |  1    1   0   1   1
'd'    |  1   -1  -1   0  -1
'e'    |  1   -1  -1   1   0
>>> cdp.exportGraphViz()
```



## On ranking from different opinions

1. On ranking from different opinions
  - Definition of the ranking problem
  - Linear Rankings
  - Majority margins
2. Types of ranking rules
  - Borda type rules
  - Condorcet : Ranking-by-choosing rules
  - Condorcet : Ranking-by-scoring rules
3. A classification of ranking rules
  - Condorcet-consistency
  - M-ordinality and M-invariance
  - Which ranking rule should we use ?

## Ranking rule

- A *profile*  $u = \{R_1, R_2, \dots, R_q\}$  is a list of  $q$  linear rankings.
- This profile  $u$  is the input of a **ranking rule** :  $u \rightarrow f(u)$ .
- The output of a ranking rule can be :
  - one (*SLR*) or several (*MLR*) *linear* rankings ;
  - one (*SWR*) or several (*MWR*) *weak* rankings (with ties).
- We present hereafter three types of ranking rules :
  1. *Rank analysis* based ranking-by-scoring rules (**Borda** type) ;
  2. *Pairwise majority margins* based rules (**Condorcet** type) :
    - 2.1 Ranking-by-choosing rules ;
    - 2.2 Ranking-by-scoring rules.

## Borda's candidate-to-rank matrix

The **candidate-to-rank matrix**  $Q_{ij}$  counts the number of times the candidate  $a_i$  is ranked at position  $j$ .

$$Q_{ij} = \{ \# \text{ rankings : } a_i \text{ is ranked at the } j \text{ th position} \}$$

Borda rank analysis tableau				Q <sub>-ij</sub>				
voter's weight	marginal ranking	candidates		1	2	3	4	5
8	acbed	'a'		8	0	0	0	17
7	ebcda	'b'		4	7	14	0	0
4	dcbea	'c'		2	12	7	4	0
4	bdeca	'd'		4	6	0	7	8
2	cdbea	'e'		7	0	4	14	0

## Borda's rule

- A **Borda score**  $B$  is computed for each candidate  $a_i$  as follows :

$$B(a_i) = \sum_{j=1}^n (Q_{ij} \times j)$$

The candidates are ranked from the lowest to the largest according to the Borda scores (to be mimized).

- A generalization of the Borda rule is to use any set of weights representing the ranks. Let  $w_1 < w_2 < \dots < w_n$  be increasing weights of the ranks. Then the Borda scores  $B$  are defined as follows :

$$B(a_i) = \sum_{j=1}^n (Q_{ij} \times w_j)$$

- The **Borda ranking**  $\succeq_B$  is the weak ranking defined as follows :

$$\forall x, y \in X, (x, y) \in \succeq_B \Leftrightarrow b_x \leq b_y.$$

## Example (Borda's weighted scores)

Borda rank analysis tableau							
candi- dates	candidate x rank					Borda scores	
	1	2	3	4	5	w1	w2
'b'	4	7	14	0	0	60	88
'c'	2	12	7	4	0	63	85
'e'	7	0	4	14	0	75	111
'd'	4	6	0	7	8	84	122
'a'	8	0	0	0	17	93	144
w1	1	2	3	4	5		
w2	1	2	5	6	8		

We observe two different rankings :  $R_{w1} : bceda$  and  $R_{w2} : cbeda$ , depending hence on the actual rank weights. Notice that the original Borda ranking  $R_{w1}$  is not consistent with the majority relation, which is  $R_{w2}$ . Given the ordinal nature of the input data, there is no information on how to assign weights to the ranks.

## Generalized ranks-based rules

**Definition (Borda type Rules, SWR/candidate×rank analysis)**

Let  $r_{ik}$ ,  $i = 1..n$ ,  $k = 1..q$  be the rank of candidate  $a_i$  in ranking  $R_k$ , and  $w_1, w_2, \dots, w_q$  be a set of given rank weights. We may rank :

- according to the average weighted rank :

$$B(a_i) = \frac{1}{q} \sum_{k=1}^q (r_{ik} \times w_k)$$

- according to the weighted median rank :

$$B(a_i) = \text{median}[(r_{i1} \times w_1), (r_{i2} \times w_2), \dots, (r_{iq} \times w_q)]$$

- by minimizing a given distance function (Cook & Seiford).

## Condorcet : Ranking-by-choosing Rules

**Definition (Kohler's Rule, MLR/majority margins  $M(x, y)$ )**

*Optimistic sequential maximin rule.* At step  $r$  (where  $r$  goes from 1 to  $n$ ) :

- Compute for each candidate  $x$  the smallest  $M(x, y)$  ( $x \neq y$ ) ;
- Select the candidate for which this minimum is maximal. If there are ties select in lexicographic order ;
- Put the selected candidate at rank  $r$  in the final ranking ;
- Delete the row and the column corresponding to the selected candidate and restart from (1).

## Example (Kohler's ranking rule)

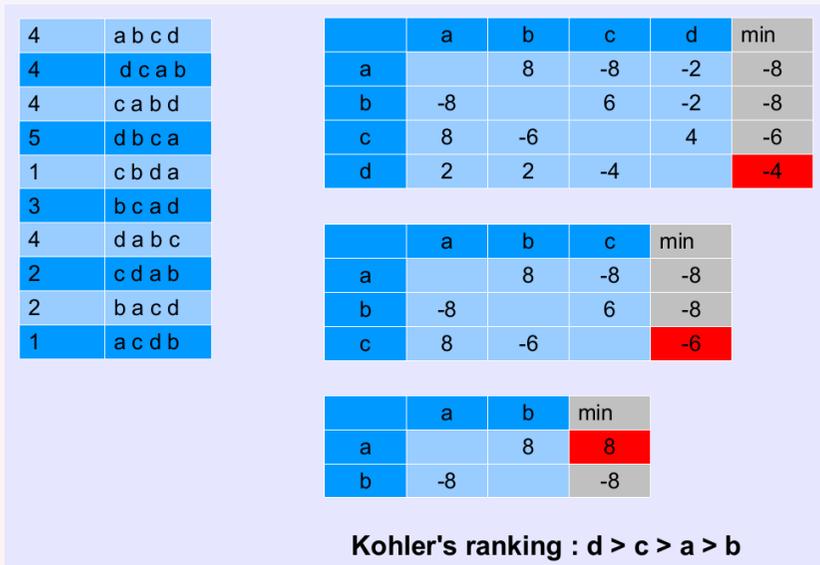


FIGURE – 2. Source : Cl. Lamboray

## Ranking-by-choosing Rules – continue

Definition (**Arrow & Raynaud's Rule**, *MLR*/majority margins  $M(x, y)$ )

*Pessimistic* (prudent) sequential **minmax rule**. At step  $r$  (where  $r$  goes from 1 to  $n$ ) :

1. Compute for each candidate  $x$  the largest  $M(x, y)$  ( $x \neq y$ ) ;
2. Select the candidate for which this maximum is minimal. If there are ties select the candidates in lexicographic order ;
3. Put the selected candidate at rank  $n - r + 1$  in the final ranking ;
4. Delete the row and the column corresponding to the selected candidate and restart from (1).

## Ranking-by-choosing Rules – continue

Definition (**Ranked Pairs' Rule**, *MLR*/majority margins  $M(x, y)$ )

1. Rank in decreasing order the ordered pairs  $(x, y)$  of candidates according to their majority margin  $M(x, y)$ .
2. Take any linear ranking compatible with this weak order.
3. Consider the pairs  $(x, y)$  in that order and do the following :
  - 3.1 If the considered pair creates a cycle with the already blocked pairs, skip this pair ;
  - 3.2 If the considered pair does not create a cycle with the already blocked pairs, block this pair.

## Example (Ranked Pairs rule)

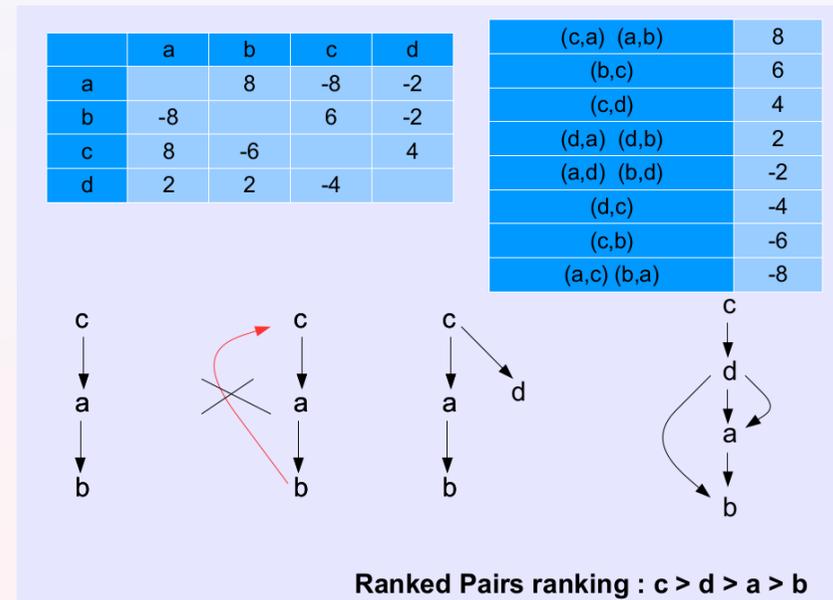


FIGURE – 3. Source : Cl. Lamboray

### Example (Condorcet : Ranking-by-choosing)

```
>>> from linearOrders import *
>>> ko = KohlerOrder(cd)
>>> ko.kohlerRanking
['c', 'b', 'e', 'd', 'a']
>>> cdcd = ~(-cd) # codual of cd
>>> ar = KohlerOrder(cdcd) # Arrow-Raynaud rule
>>> ar.kohlerRanking
['c', 'b', 'e', 'd', 'a']
>>> rp = RankedPairsOrder(cd)
>>> rp.rankedPairsRanking
['c', 'b', 'e', 'd', 'a']
```

Kohler's, Arrow&Raynaud's and the RankedPairs rule all result in the same unique linear ranking : 'cbeda', which corresponds to the majority relation C.

### Condorcet : Ranking-by-scoring rules

#### Definition (NetFlows Rule, MWR/majority margins $M(x, y)$ )

- The idea is that the more a given candidate beats other candidates the better it is.
- Similarly, the more other candidates beat a given candidate, the lower this candidate should be ranked.
- The **NetFlows score**  $n_x$  of candidate  $x$  is defined as follows :

$$n_x = \sum_y [M(x, y) - M(y, x)].^1$$

- The **NetFlows ranking**  $\succeq_N$  is the weak ranking defined as follows :  $\forall x, y \in X, (x, y) \in \succeq_N \Leftrightarrow n_x \geq n_y$ .

1. Notice that in the case of linear profiles, we may drop the  $-M(y, x)$  term due to the zero sum property.

### Condorcet : Ranking-by-scoring rules

#### Definition (Copeland's Rule, MWR/majority relation $C(x, y)$ )

- The idea is that the more a given candidate beats other candidates at majority the better it should be ranked.
- Similarly, the more other candidates beat a given candidate at majority, the lower this candidate should be ranked.
- The **Copeland score**  $c_x$  of candidate  $x$  is defined as follows :

$$\begin{aligned} c_x &= \#\{y \neq x \in X : M(x, y) > 0\} \\ &\quad - \#\{y \neq x \in X : M(y, x) > 0\} \\ &= \sum_y (C(x, y) - C(y, x)). \end{aligned}$$

- The **Copeland ranking**  $\succeq_C$  is the weak ranking defined as follows :  $\forall x, y \in X, (x, y) \in \succeq_C \Leftrightarrow c_x \geq c_y$ .

### Condorcet : Ranking-by-scoring rules

#### Definition (Kemeny's Rule, MLR/majority margins $M(x, y)$ )

- The idea is finding a compromise ranking  $R$  that minimizes the distance to the  $q$  marginal linear rankings of the voting profile according to the symmetric difference measure :  $\delta$ . If  $R_1$  and  $R_2$  are two relations,  $\delta(R_1, R_2) = |R_1 \oplus R_2| / 2$ .
- The **Kemeny ranking**, also called *median* ranking,  $R^*$  is a solution of the following optimization problem :

$$\min_{arg_R} \delta(M, R) \quad \equiv \quad \max_{arg_R} \sum_{(x,y) \in R} [M(x, y) \times r(x, y)]$$

such that  $R$  is a linear ranking.

- The distance  $\delta(M, R^*)$  is called the **Kemeny index** of a preference profile. Computing the Kemeny index is an NP-complete problem and Kemeny rankings are generally not unique.

## Condorcet : Ranking-by-scoring rules

### Definition (Slater's Rule, MLR/majority relation $C(x, y)$ )

- The idea is to select a ranking that is closest according to the symmetric difference distance  $\delta$  to the Condorcet digraph's polarized relation  $M_{>0}$ .
- The Slater ranking  $R^*$  is a solution of the following optimization problem :

$$\text{minarg}_R \delta(M_{>0}, R) \equiv \text{maxarg}_R \sum_{(x,y) \in R} [C(x,y) \times r(xRy)]$$

such that  $R$  is a linear ranking.

- The distance  $\delta(R^*, M_{>0})$  is called the Slater index of a preference profile. Computing the Slater index of a profile is an NP-hard problem and Slater rankings are even less unique than Kemeny rankings.

## Example (Condorcet : ranking-by-scoring)

```
* ---- Majority margins ----
Mxy | 'a' 'b' 'c' 'd' 'e'
'a' |  0  -9  -9  -9  -9
'b' |  9   0  -3  13  11
'c' |  9   3   0   9   3
'd' |  9  -13  -9   0  -5
'e' |  9  -11  -3   5   0

>>> cd.computeNetFlowsRanking(Debug=True)
OrderedDict([('b', 60), ('c', 48), ('e', 0), ('d', -36), ('a', -72.0)])
['b', 'c', 'e', 'd', 'a']
>>> cd.computeCopelandRanking(Debug=True)
OrderedDict([('c', 4), ('b', 2), ('e', 0), ('d', -2), ('a', -4)])
['c', 'b', 'e', 'd', 'a']
>>> from linearOrders import KemenyOrder
>>> ke = KemenyOrder(cd); ke.maximalRankings
[['c', 'b', 'e', 'd', 'a']]
>>> kecd = KemenyOrder(cdcd); kecd.maximalRankings
[['c', 'b', 'e', 'd', 'a']]
```

The NetFlows rule, like the Borda rule, inverts the two top ranked candidates : 'bceda', whereas Copeland's, Kemeny's and Slater's rules result again in the same unique ranking : 'cbeda'.

## Content Lecture 3

- On ranking from different opinions
  - Definition of the ranking problem
  - Linear Rankings
  - Majority margins
- Types of ranking rules
  - Borda type rules
  - Condorcet : Ranking-by-choosing rules
  - Condorcet : Ranking-by-scoring rules
- A classification of ranking rules
  - Condorcet-consistency
  - M-ordinality and M-invariance
  - Which ranking rule should we use ?

## A classification of ranking rules

### Definition (Condorcet-consistency)

A ranking rule is **Condorcet-consistent** if the following holds :  
*If the majority relation is a linear ranking, then this ranking is the unique solution of the ranking rule.*

### Property (Condorcet consistent rules)

*Kemeny's, Slater's, Copeland's, Kohler's and the RankedPairs rule are all Condorcet-consistent. The Borda and the NetFlows rules are, both, not Condorcet-consistent.*

## A classification of ranking rules

### Definition (*M*-ordinality)

A ranking rule is *M*-**ordinal** if its ranking result only depends on the order of the majority margins.

### Property (*M*-ordinal rules)

Slater's, Copeland's, Kohler's and the RankedPairs rule are all *M*-ordinal. The Kemeny and the NetFlows rules are not *M*-ordinal.

### Definition (*M*-invariance)

A ranking rule is *M*-**invariant** if its ranking result only depends on the sign of the majority margins.

### Property (*M*-invariant rules)

Slater's and Copeland's rule are both *M*-invariant. Kohler's and the RankedPairs rules are not *M*-invariant.

## A classification of ranking rules by Cl. Lamboray

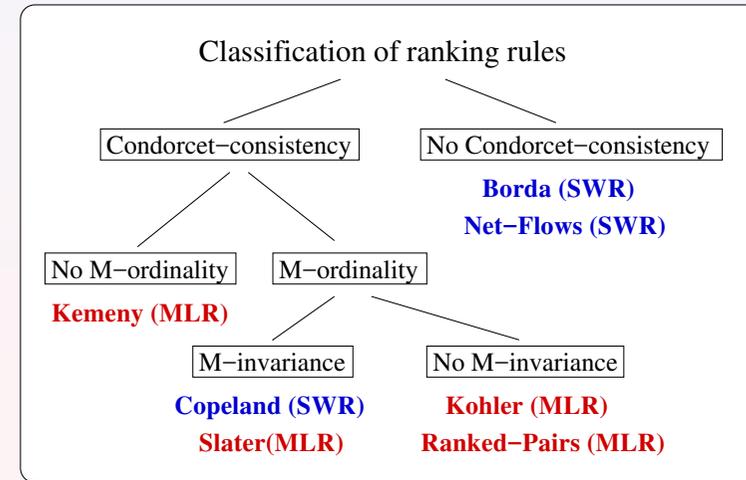


FIGURE – 4. SWR : single weak ranking, MLR : multiple linear rankings

## Which ranking rule should we use

- There is no perfect ranking rule (cf Arrow's theorem).
- What properties of a ranking rule are useful or required ?
- Axiomatic characterizations of the ranking rules.
- More or less consensual global rankings ?
- Correlation with the majority margins  $M(x, y)$  ?
- Fitness for big data : computational complexity ?

### Example (Which is the *better* social ranking ?)

```
>>> from votingProfiles import *
>>> v = LinearVotingProfile('example1')
>>> v.showHTMLVotingHeatmap(rankingRule='Copeland')
>>> v.showHTMLVotingHeatmap(rankingRule='NetFlows')
```

criteria	v5	v3	v2	v4	v1
weights	2	4	7	4	8
tau(*)	0.60	0.40	0.40	0.20	0.20
c	5	4	3	2	4
b	3	3	4	5	3
e	2	2	5	3	2
d	4	5	2	4	1
a	1	1	1	1	5

criteria	v2	v5	v4	v3	v1
weights	7	2	4	4	8
tau(*)	0.60	0.40	0.40	0.20	0.00
b	4	3	5	3	3
c	3	5	2	4	4
e	5	2	3	2	2
d	2	4	4	5	1
a	1	1	1	1	5

Figure – 5. Copeland – versus NetFlows ranking.

(\*) tau : Ordinal (Kendall) correlation between *marginal* and *global* ranking. The ranks are of *reversed Borda* type :  $w_1 = 5, w_2 = 4, w_3 = 3, w_4 = 2, w_5 = 1$ .

## Example (Correlations with the majority margins)

```
>>> cd.recodeValuation(1,1) # normalizing the majority margins
>>> from linearOrders import CopelandOrder, NetFlowsOrder
>>> cop = CopelandOrder(cd); cop.copelandOrder
['a', 'd', 'e', 'b', 'c']
>>> corr = cd.computeOrderCorrelation(cop.copelandOrder)
>>> cd.showCorrelation(corr)
Correlation indexes:
Crisp ordinal correlation : +1.000
Valued equivalence       : +0.320
Epistemic determination  : 0.320
>>> nf = NetFlowsOrder(cd); nf.netFlowsOrder
['a', 'd', 'e', 'c', 'b']
>>> corr cd.computeOrderCorrelation(nf.netFlowsOrder)
Correlation indexes:
Crisp ordinal correlation : +0.925
Valued equivalence       : +0.296
Epistemic determination  : 0.320
```

In this example, the *Condorcet-consistency* property assures that the *Copeland*, *Kemeny* and *Slater* ranking rules all deliver a perfectly matching ordinal result ( $\tau = +1.0$ ), whereas the *Net-Flows* rule inverts the top candidates ( $\tau = +0.925$ ). The epistemic determination of the majority margins is 0.32, ie the ordinal correlations are supported here in average by a  $(1.0 + 0.32)/2 = 66\%$  majority, ie 16/25 voters.

On ranking from different opinions  
○○  
○○  
○○○

Types of ranking rules  
○○  
○○○○  
○○○○○○  
○○○○○

A classification of ranking rules  
○  
○  
○○  
○○○○●

## Exercise (Claude Lamboray, PhD thesis p. 35)

Apply all the previous ranking rules on the following profile of 10 weighted linear orders defined on 4 candidates  $\{a, b, c, d\}$  as shown below; discuss the results.

4 : abcd	3 : bcad
4 : dcab	4 : dabc
4 : cabd	2 : cdab
5 : dbca	2 : bacd
1 : cbda	1 : acdb

34 / 36

On ranking from different opinions  
○○  
○○  
○○○

Types of ranking rules  
○○  
○○○○  
○○○○○○  
○○○○○

A classification of ranking rules  
○  
○  
○○  
○○○○●

On ranking from different opinions  
○○  
○○  
○○○

Types of ranking rules  
○○  
○○○○  
○○○○○○  
○○○○○

A classification of ranking rules  
○  
○  
○○  
○○○○●

## Digraph3 software resources

### Exercise (*votingProfiles* module extension)

Suppose that some voters will not provide a complete linear ranking of all the candidates. Develop Python code based on the *votingProfiles* module, that implements all the previously defined ranking rules and renders a corresponding ranking when given a *LinearVotingProfile* instance with partial ballots.

- Documentation index :  
<https://digraph3.readthedocs.io/en/latest/index.html>
- Tutorials :  
<https://digraph3.readthedocs.io/en/latest/tutorial.html>
- Reference manual :  
<https://digraph3.readthedocs.io/en/latest/techDoc.html>
- Advanced topics :  
<https://digraph3.readthedocs.io/en/latest/pearls.html>