Lecture 9: On ranking from valued pairwise outrankings

MICS: Algorithmic Decision Theory

Raymond Bisdorff

University of Luxembourg

May 4, 2020

Content

Bipolar characteristic function r

- $X = \{x, y, z, ...\}$ is a finite set of m decision alternatives;
- We define a binary relation R on X with the help of a bipolar characteristic function r taking values in the rational interval $[-1.0; 1.0]$.
- **Bipolar semantics**: For any pair $(x, y) \in X^2$,
  1. $r(x \,\mathrm{R}\, y) = +1.0$ means $x \,\mathrm{R}\, y$ valid for sure,
  2. $r(x \,\mathrm{R}\, y) > 0.0$ means $x \,\mathrm{R}\, y$ more or less valid,
  3. $r(x \,\mathrm{R}\, y) = 0.0$ means both $x \,\mathrm{R}\, y$ and $x \not\mathrm{R}\, y$ indeterminate,
  4. $r(x \,\mathrm{R}\, y) < 0.0$ means $x \not\mathrm{R}\, y$ more or less valid,
  5. $r(x \,\mathrm{R}\, y) = -1.0$ means $x \not\mathrm{R}\, y$ valid for sure.
- **Boolean operations**: Let ϕ and ψ be two relational propositions.
  1. $r(\neg\phi) = -r(\phi)$.
  2. $r(\phi \vee \psi) = \max\big(r(\phi), r(\psi)\big)$,
  3. $r(\phi \wedge \psi) = \min\big(r(\phi), r(\psi)\big)$.

Weakly complete binary relations

Let R be an r-valued binary relation defined on X.

Definition

We say that R is weakly complete on X if, for all $(x, y) \in X^2$, **either** $r(x \, \mathrm{R} \, y) \geqslant 0.0$ **or** $r(y \, \mathrm{R} \, x) \geqslant 0.0$.

Examples

1. Marginal semi-orders (orders with discrimination thresholds) observed on each criterion,
2. Global weighted "*at least as performing as*" relations,
3. Outranking relations (polarized with considerable performance differences),
4. Fusion of (vague) weak or linear orders,
5. Ranking-by-choosing ordering results.

Universal properties

Let $\mathcal{R}$ denote the set of all possible weakly complete relations definable on X.

Property ($\mathcal{R}$-internal operations)

1. *The convex combination of any finite set of such weakly complete relations remains a weakly complete relation.*
2. *The disjunctive combination of any finite set of such weakly complete relations remains a weakly complete relation.*
3. *The epistemic-conjunctive (resp. -disjunctive) combination of any finite set of such weakly complete relations remains a weakly complete relation.*

Examples: Concordance of linear-, weak- or semi-orders, bipolar outranking (concordance-discordance) relations.

Useful properties

We say that a binary relation $R \in \mathcal{R}$ verifies the *coduality principle* ($> \equiv \not\leqslant$), if the converse of its negation equals its asymetric part : $\min\left(r(x \, \mathrm{R} \, y), -r(y \, \mathrm{R} \, x)\right) \; = \; -r(y \, \mathrm{R} \, x)$. Let $\mathcal{R}^{cd}$ denote the set of all possible relations $R \in \mathcal{R}$ that verify the coduality principle.

Property (Coduality principle)

The convex and epistemic-disjunctive (resp. -conjunctive) combinations of a finite set of relations in $\mathcal{R}^{cd}$ verify again the coduality principle.

Examples: Marginal linear-, weak- and semi-orders; concordance and bipolar outranking relations; all, verify the coduality principle.

The Multiple Criteria Ranking Problem

- A ranking problem traditionally consists in the search for a linear ranking (without ties) or a weak ranking (with ties) of the set of alternatives.
- A paricular ranking is computed with the help of a ranking rule which aggregates preferences over all decision makers and/or performance criteria into a global (weak) ranking based on (pairwise) bipolar-valued outranking characteristics $r(\succsim)$.
- Characteristic properties of ranking rules:
  1. A ranking rule is called Condorcet-consistent when the following holds: If the median cut relation (Condorcet majority) is a linear ranking, then this linear ranking is the unique solution of the ranking rule;
  2. A ranking rule is called r-ordinal if its result only depends on the order of the bipolar outranking characteristics $r(\succsim)$;
  3. A ranking rule is called r-invariant if its result only depends on the sign of the bipolar outranking characteristics $sign(r(\succsim))$.

A classification of ranking rules by Cl. Lamboray

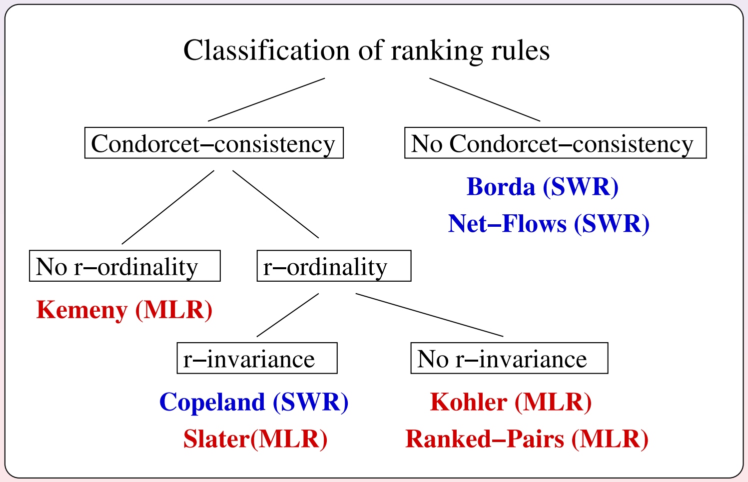

Figure: 4. *SWR*: single weak ranking, *MLR*: multiple linear rankings

Quality criteria for ranking results

- Best satisfying Condorcet consistency: Highest possible ordinal correlation with global outranking relation $\succsim$.
- Best majority significance supported: Highest possible mean weighted marginal correlation.
- Best multiple criteria compromise: Lowest possible standard deviation of the mean marginal correlation.
- Fairest ranking result: Highest mean marginal correlation minus one standard deviation.

Ranking-by-scoring Rule – I

Definition (Copeland's Rule, WO/Condorcet Digraph)

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

$$\begin{aligned} c_x \quad = & \quad |\{y \neq x \in X : r(x \succsim y) > 0\}| \\ - & \quad |\{y \neq x \in X : r(y \succsim x) > 0\}| \end{aligned}$$

- The Copeland ranking $\succeq_C$ is the weak order defined as follows: $\forall x, y \in X, \quad (x, y) \in \succeq_C \quad \Leftrightarrow \quad c_x \geqslant c_y$.
- Copelande's rule is invariant under the codual transform.

Random Cost-Benefit performance tableau

```
>>> from randomPerfTabs import\
     RandomCBPerformanceTableau
>>> t = RandomCBPerformanceTableau(
     numberOfActions=7,numberOfCriteria=5,
     seed=100)
>>> t
 *--- instance description ------*
 Instance class:
       RandomCBPerformanceTableau
 Seed : 100
 Instance name : randomCBperftab
 # Actions     : 7
 # Objectives  : 2
 # Criteria    : 5
>>> t.showHTMLPerformanceTableau()
```

criteria	b1	b2	c1	c2	c3
weight	3.00	3.00	2.00	2.00	2.00
a1c	2.00	20.66	-28.65	-79.26	-3.19
a2n	3.00	47.34	-15.47	-52.85	-44.47
a3n	3.00	57.44	-81.69	-77.77	-66.35
a4c	2.00	69.06	-77.23	-57.77	-34.80
a5a	7.00	35.23	-57.33	-39.65	-43.45
a6n	9.00	52.77	-39.33	-58.08	-71.82
a7a	5.00	92.47	-79.57	-34.87	-87.54

We observe two benefit criteria (b1,b2) of signifinace 3.0 and three costs criteria (c1,c2,c3) of significance 3.0. There are two advantageous (a5a,a7a), two cheap (a1c,a4c) and three neutral (a2n,a3n,a6n) decision alternatives.

Computing a Copeland ranking

```
>>> from outrankinDigraph import *
>>> g = BipolarOutrankingDigraph(g,
                Normalized=True)
>>> from linearOrders\
        import CopelandRanking
>>> cop = CopelandRanking(g)
>>> cop.showRanking()
 ['a6','a5','a2','a7','a4','a1','a3']
>>> corr =\
      g.computeOrdinalCorrelation(cop)
>>> g.showCorrelation(corr)
 Correlation indexes:
  Crisp ordinal correlation  : +0.728
  Epistemic determination    :  0.335
  Bipolar-valued equivalence : +0.244
>>> t.showHTMLPerformanceHeatmap(
     pageTitle='Copeland Ranking',
     actionsList=cop.copelandRanking,
     colorLevels=5,Correlations=True)
```

criteria	b1	c2	c1	b2	c3
weights	+3.00	+2.00	+2.00	+3.00	+2.00
tau$^{(*)}$	+0.62	+0.40	+0.31	-0.02	-0.26
a6n	9.00	-58.08	-39.33	52.77	-71.82
a5a	7.00	-39.65	-57.33	35.23	-43.45
a2n	3.00	-52.85	-15.47	47.34	-44.47
a7a	5.00	-34.87	-79.57	92.47	-87.54
a4c	2.00	-57.77	-77.23	69.06	-34.80
a1c	2.00	-79.26	-28.65	20.66	-3.19
a3n	3.00	-77.77	-81.69	57.44	-66.35

Color legend:

quantile	20.00%	40.00%	60.00%	80.00%	100.00%

```
>>> t.showRankingConsensusQuality(
             cop.copelandRanking)
Summary:
mean marg. correlation (a): +0.224
Standard deviation (b)    : +0.317
Ranking fairness (a)-(b)  : -0.093
```


Ranking-by-scoring Rule – II

Definition (Net-flows's Rule, WO/$r(\succsim)$)

- The idea is that the more a given alternative beats other alternatives the better it is. Similarly, the more other alternatives beat a given alternative, the lower this alternative should be ranked.
- The net-flows score c_x of alternative $x \in X$ is defined as follows:

$$n_x \;=\; \sum_{y \in X \,\wedge\, y \neq x} \big[\, r(x \succsim y) - r(y \succsim x) \,\big]$$

- The net-flows ranking $\succeq_C$ is the weak order defined as follows: $\forall x, y \in X, \quad (x, y) \in \succeq_C \quad \Leftrightarrow \quad n_x \geqslant n_y.$
- The NetFlows'rule is invariant under the codual transform. The rule gives the same result for both the $r(\succsim)$ and the $r(\not\precsim)$ digraph.

Computing a NetFlows ranking

```
>>> # same performance tableau t
>>> from outrankingDigraphs import\
       BipolarOutrankingDigraph
>>> g = BipolarOutrankingDigraph(t,
              Normalized=True)
>>> from linearOrders\
          import NetFlowsRanking
>>> nf = NetFlowsRanking(g)
>>> nf.showRanking()
 ['a2','a6','a7','a5','a4','a1','a3']
>>> corr =\
      g.computeOrdinalCorrelation(nf)
>>> g.showCorrelation(corr)
 Correlation indexes:
  Crisp ordinal correlation  : +0.716
  Epistemic determination    :  0.335
  Bipolar-valued equivalence : +0.240
>>> t.showHTMLPerformanceHeatmap(
       pageTitle = 'NetFlows Ranking',
      colorLevels=5,Correlations=True)
>>> # NetFlows rule is default
```

criteria	c2	c1	b1	b2	c3
weights	+2.00	+2.00	+3.00	+3.00	+2.00
tau$^{(*)}$	+0.50	+0.40	+0.33	+0.12	-0.26
a2n	-52.85	-15.47	3.00	47.34	-44.47
a6n	-58.08	-39.33	9.00	52.77	-71.82
a7a	-34.87	-79.57	5.00	92.47	-87.54
a5a	-39.65	-57.33	7.00	35.23	-43.45
a4c	-57.77	-77.23	2.00	69.06	-34.80
a1c	-79.26	-28.65	2.00	20.66	-3.19
a3n	-77.77	-81.69	3.00	57.44	-66.35

Color legend:

quantile	20.00%	40.00%	60.00%	80.00%	100.00%

```
>>> t.showRankingConsensusQuality(
             nf.netFlowsRanking)
Summary:
mean marg. correlation (a): +0.220
Standard deviation (b)    : +0.251
Ranking fairness (a)-(b)  : -0.031
```

Ranking-by-scoring Rule – III

Definition (Kemeny's Rule, SLO/$r(\succsim)$)

- The original idea is finding a compromise ranking O that minimizes the distance to the q marginal linear orders of the voting profile according to the symmetric difference measure.
- With bipolar-valued outranking digraphs, the Kemeny (also called *median*) order O^* is a solution of the following optimization problem:

$$\begin{array}{ll} \text{maxarg}_O & \sum_{(x,y)\in O} \big(r(x \succsim y) \; - \; r(y \succsim x)\big) \\ \text{such that} & O \text{ is a linear order on X} \end{array}$$

- Finding a Kemeny order O^* is an NP-complete problem. We need to inspect all possible permutations of the decision alternatives.
- The Kemeny rule is invariant under the codual transform.

Computing a Kemeny ranking

```
>>> # same outranking digraph g
>>> from linearOrders\
        import KemenyRanking
>>> ke = KemenyRanking(g)
>>> ke.showRanking()
 ['a6','a5','a7','a2','a4','a3','a1']
>>> corr =\
      g.computeOrdinalCorrelation(ke)
>>> g.showCorrelation(corr)
 Correlation indexes:
  Crisp ordinal correlation  : +0.893
  Epistemic determination    :  0.335
  Bipolar-valued equivalence : +0.300
>>> t.showHTMLPerformanceHeatmap(
     pageTitle='Kemeny Ranking',
     actionsList=ke.kemenyRanking,
     colorLevels=5,Correlations=True)
```

criteria	b1	c2	b2	c1	c3
weights	+3.00	+2.00	+3.00	+2.00	+2.00
tau(*)	+0.81	+0.55	+0.17	+0.12	-0.45
a6n	9.00	-58.08	52.77	-39.33	-71.82
a5a	7.00	-39.65	35.23	-57.33	-43.45
a7a	5.00	-34.87	92.47	-79.57	-87.54
a2n	3.00	-52.85	47.34	-15.47	-44.47
a4c	2.00	-57.77	69.06	-77.23	-34.80
a3n	3.00	-77.77	57.44	-81.69	-66.35
a1c	2.00	-79.26	20.66	-28.65	-3.19

Color legend:

quantile	20.00%	40.00%	60.00%	80.00%	100.00%

```
>>> t.showRankingConsensusQuality(
            ke.kemenyRanking)
Summary:
mean marg. correlation (a): +0.280
Standard deviation (b)    : +0.423
Ranking fairness (a)-(b)  : -0.143
```


Ranking-by-scoring Rule – IV

Definition (Slater's Rule, SLO/Condorcet Digraph)

- The idea is to select a ranking that is closest according to the symmetric difference distance to the Condorcet digraph's relation $r(x \succsim y)_{>0}$.
- The Slater order O^* is a solution of the following optimization problem:

$$\begin{array}{ll} \text{maxarg}_O & \sum_{(x,y)\in O} \big(r(x \succsim y)_{>0} \; - \; r(y \succsim x)_{>0}\big) \\ \text{such that} & O \text{ is a linear order on X} \end{array}$$

- Slater's rule is again invariant under the codual transform and an NP-hard problem.

Computing a Slater ranking

```
>>> # same outranking digraph g
>>> from linearOrders\
        import SlaterRanking
>>> sl = SlaterRanking(g)
>>> sl.showRanking()
 ['a6','a5','a7','a2','a4','a3','a1']
 # same as the Kemeny ranking
>>> corr =\
      g.computeOrdinalCorrelation(sl)
>>> g.showCorrelation(corr)
 Correlation indexes:
  Crisp ordinal correlation  : +0.893
  Epistemic determination    :  0.335
  Bipolar-valued equivalence : +0.300
>>> t.showHTMLPerformanceHeatmap(
     pageTitle='Slater Ranking',
     actionsList=sl.slaterRanking,
     colorLevels=5,Correlations=True)
```

criteria	b1	c2	b2	c1	c3
weights	+3.00	+2.00	+3.00	+2.00	+2.00
tau(*)	+0.81	+0.55	+0.17	+0.12	-0.45
a6n	9.00	-58.08	52.77	-39.33	-71.82
a5a	7.00	-39.65	35.23	-57.33	-43.45
a7a	5.00	-34.87	92.47	-79.57	-87.54
a2n	3.00	-52.85	47.34	-15.47	-44.47
a4c	2.00	-57.77	69.06	-77.23	-34.80
a3n	3.00	-77.77	57.44	-81.69	-66.35
a1c	2.00	-79.26	20.66	-28.65	-3.19

Color legend:

quantile	20.00%	40.00%	60.00%	80.00%	100.00%

```
>>> t.showRankingConsensusQuality(
            sl.slaterRanking)
Summary:
mean marg. correlation (a): +0.280
Standard deviation (b)    : +0.423
Ranking fairness (a)-(b)  : -0.143
```

Ranking-by-choosing Rule – I

Definition (Kohler's Rule, SLO/$r(\succsim)$)

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

1. Compute for each alternative x the smallest $r(x \succsim y)$ $(x \neq y)$;
2. Select the alternative for which this minimum is maximal. If there are ties select the first these alternatives in lexicographic order;
3. Put the selected alternative at rank r in the final ranking;
4. Delete the row and the column corresponding to the selected alternative and restart from (1).

Computing a Kohler ranking

```
>>> # same outranking digraph g
>>> from linearOrders\
        import KohlerRanking
>>> ko = KohlerRanking(g)
>>> ko.showRanking()
 ['a6','a5','a7','a2','a4','a3','a1']
 # same as the Kemeny ranking
>>> corr =\
      g.computeOrdinalCorrelation(ko)
>>> g.showCorrelation(corr)
 Correlation indexes:
  Crisp ordinal correlation  : +0.893
  Epistemic determination    :  0.335
  Bipolar-valued equivalence : +0.300
>>> t.showHTMLPerformanceHeatmap(
     pageTitle='Kohler Ranking',
     actionsList=ko.kohlerRanking,
     colorLevels=5,Correlations=True)
```

criteria	b1	c2	b2	c1	c3
weights	+3.00	+2.00	+3.00	+2.00	+2.00
tau(*)	+0.81	+0.55	+0.17	+0.12	-0.45
a6n	9.00	-58.08	52.77	-39.33	-71.82
a5a	7.00	-39.65	35.23	-57.33	-43.45
a7a	5.00	-34.87	92.47	-79.57	-87.54
a2n	3.00	-52.85	47.34	-15.47	-44.47
a4c	2.00	-57.77	69.06	-77.23	-34.80
a3n	3.00	-77.77	57.44	-81.69	-66.35
a1c	2.00	-79.26	20.66	-28.65	-3.19

Color legend:

quantile	20.00%	40.00%	60.00%	80.00%	100.00%

```
>>> t.showRankingConsensusQuality(
            ko.kohlerRanking)
Summary:
mean marg. correlation (a): +0.280
Standard deviation (b)    : +0.423
Ranking fairness (a)-(b)  : -0.143
```


Ranking-by-choosing Rule – II

Definition (Arrow & Raynaud's Rule, SLO/majority margins $r(\succsim)$)

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

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

The Arrow & Raynaud ranking may be computed with Kohler's rule but applied to the dual transform of the outranking digraph.

Computing an Arrow & Raynaud ranking

```
>>> # same outranking digraph g
>>> from linearOrders\
        import KohlerRanking
>>> ar = KohlerRanking((-g))
>>> ar.showRanking()
 ['a7','a2','a4','a6','a5','a3','a1']
>>> corr =\
      g.computeRankingCorrelation(ar.kol
>>> g.showCorrelation(corr)
 Correlation indexes:
  Crisp ordinal correlation  : +0.787
  Epistemic determination    :  0.335
  Bipolar-valued equivalence : +0.264
>>> t.showHTMLPerformanceHeatmap(
     pageTitle='Arrow&Raynaud Ranking',
     actionsList=ar.kohlerOrder,
     colorLevels=5,Correlations=True)
```

criteria	c2	b2	b1	c1	c3
weights	+2.00	+3.00	+3.00	+2.00	+2.00
tau(*)	+0.64	+0.60	+0.24	-0.07	-0.36
a7a	-34.87	92.47	5.00	-79.57	-87.54
a2n	-52.85	47.34	3.00	-15.47	-44.47
a4c	-57.77	69.06	2.00	-77.23	-34.80
a6n	-58.08	52.77	9.00	-39.33	-71.82
a5a	-39.65	35.23	7.00	-57.33	-43.45
a3n	-77.77	57.44	3.00	-81.69	-66.35
a1c	-79.26	20.66	2.00	-28.65	-3.19

Color legend:

quantile	20.00%	40.00%	60.00%	80.00%	100.00%

```
>>> t.showRankingConsensusQuality(
            ar.kohlerOrder)
Summary:
mean marg. correlation (a): +0.244
Standard deviation (b)    : +0.366
Ranking fairness (a)-(b)  : -0.122
```


Ranking-by-choosing Rule – III

Definition (Ranked-Pairs' Rule, SLO/majority margins $r(\succsim)$)

1. Rank in decreasing order the ordered pairs (x, y) of alternatives according to $r(x \succsim y) - r(y \succsim x)$.
2. Take any linear order 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.

Computing a Ranked-Pairs ranking

```
>>> # same outranking digraph g
>>> from linearOrders import\
         RankedPairsRanking
>>> rp = RankedPairsRanking(g)
>>> rp.showRanking()
 ['a7','a2','a4','a6','a5','a3','a1']
 # same as the Arrow&Raynaud result
>>> corr =\
   g.computeOrdinalCorrelation(rp)
>>> g.showCorrelation(corr)
 Correlation indexes:
  Crisp ordinal correlation  : +0.787
  Epistemic determination    :  0.335
  Bipolar-valued equivalence : +0.264
>>> t.showHTMLPerformanceHeatmap(
    pageTitle='RankedPairs Ranking',
    actionsList=rp.rankedPaisRanking,
    colorLevels=5,Correlations=True)
```

criteria	c2	b2	b1	c1	c3
weights	+2.00	+3.00	+3.00	+2.00	+2.00
tau(*)	+0.64	+0.60	+0.24	-0.07	-0.36
a7a	-34.87	92.47	5.00	-79.57	-87.54
a2n	-52.85	47.34	3.00	-15.47	-44.47
a4c	-57.77	69.06	2.00	-77.23	-34.80
a6n	-58.08	52.77	9.00	-39.33	-71.82
a5a	-39.65	35.23	7.00	-57.33	-43.45
a3n	-77.77	57.44	3.00	-81.69	-66.35
a1c	-79.26	20.66	2.00	-28.65	-3.19

Color legend:

quantile	20.00%	40.00%	60.00%	80.00%	100.00%

```
>>> t.showRankingConsensusQuality(
             rp.rankedPairsRanking)
Summary:
mean marg. correlation (a): +0.244
Standard deviation (b)    : +0.366
Ranking fairness (a)-(b)  : -0.122
```


Summerizing all ranking results

Ranking	NetFlows	Copeland	RankedPairs(*)	Kemeny(**)
$\tau_{\succsim}$	+0.716	+0.728	+0.787	+0.893
τ_{b1}	+0.33	+0.62	+0.24	+0.81
τ_{b2}	+0.12	-0.02	+0.60	+0.17
τ_{c1}	+0.40	+0.31	-0.07	+0.12
τ_{c2}	+0.50	+0.40	+0.64	+0.55
τ_{c3}	-0.26	-0.26	-0.36	-0.45
mean(τ) (a)	+0.220	+0.224	+0.244	+0.280
stdev(τ) (b)	0.251	0.317	0.366	+0.423
fairness (a)-(b)	-0.031	-0.093	-0.122	-0.143

(*) Arrow & Raynaud's and Tideman's RankedPairs rules deliver in this didactical example a same result. This is usually not the case.

(**) Similarly, Kohler's and Slater's rules deliver the same ranking result as Kemeny's rule. This is again usually not the case.

What ranking rule should one use ? – I

1. Kemeny's rule shows, on the one hand, the most adversary ranking with highest mean marginal correlation (+0.280), but also highest marginal correlation spread (0.423), and consequently lowest fairness index (-0.143).
2. The NetFlows rule, on the other hand, shows with highest fairness index (-0.031), the most consensual ranking result. A result due, despite the lowest mean marginal correlation (+0.220), to the lowest marginal correlation spread (0.251).
3. The Copeland and the RankedPairs rules show ranking results with a quality in between both previous extremes.
4. Depending on the numerical diversity of the pairwise bipolar-valued outranking characteristics, the NetFlows rule, by tempering the Condorcet consistency (the potential dictatorship of the majority principle), gives usually the fairest (most consensual) ranking result.

What ranking rule should one use ? – II

1. Kemeny's and Slater's ranking-by-scoring rules, besides potentially delivering multiple weak rankings, are furthermore computationally difficult problems and exact ranking results are only computable for tiny outranking digraphs (order < 20).
2. Similarly, the ranking-by-choosing and their dual, the ordering-by-choosing rules, are unfortunately not scalable to outranking digraphs of larger orders (> 100).
3. Only Copeland's and the NetFlows ranking rules, with a polynomial complexity $\mathcal{O}(n^2)$, where n is the order of the outranking digraph, remain scalable for outranking digraphs with several hundred or thousand decision alternatives.

See the Digraph3 tutorial on *Ranking with multiple incommensurable criteria* (https://digraph3.readthedocs.io/en/latest/index.html).