Rank Ordering Methods for Multi-criteria Decisions

Mats Danielson and Love Ekenberg

Dept. of Computer and Systems Sciences, Stockholm University,
Forum 100, SE-164 40 Kista, Sweden
{mad,lovek}@dsv.su.se

Abstract. Criteria weights are typically cognitively demanding to elicit and numeric precision is problematic since information in real-life multi-criteria decision making often is imprecise. One important class of methods rank the criteria and receive a criteria ordering which can be handled in various ways by, e.g., converting the resulting ranking into numerical weights, so called surrogate weights. In this article, we analyse the relevance of these methods and to what extent some validation processes are strongly dependent on the simulation assumptions. We also suggest more robust methods as candidates for modelling and analysing multi-criteria decision problems of this kind.

Keywords: Multi-criteria decision analysis, Criteria weights, Criteria ranking, Rank order.

1 Introduction

Methods attempting to elicit precise criteria weights range from direct rating (DR) and point allocation (PA) methods to more elaborated ones, but when it comes to providing reasonable weights, we have significant difficulties due to the fact that we do not seem to have the required granulation capacity and we also suffer from other deficiencies. To somewhat facilitate eliciting weights from decision-makers, some of the approaches utilise ordinal or imprecise importance information to determine criteria weights and sometimes values of alternatives.[1]

[1] introduced a process utilising systematic simulations. The basic idea is to generate surrogate weights as well as "true" reference weights from some underlying distribution and investigate how well the result of using surrogate numbers match the result of using the "true" results. The idea in itself is good, but the methodology is vulnerable since the validation result is heavily dependent on the distribution used for generating the weight vectors. This article discusses some important aspects and shortcomings of some popular weight methods as well as the validation techniques for these. We also discuss the relevance and correctness of some common measurements for method validation and conclude with a discussion of more robust methods that might be better candidates.

[1] Other approaches use intervals to express uncertainty inherent in elicitation procedures, e.g., [2,3].

P. Zaraté, G.E. Kersten, and J.E. Hernández (Eds.): GDN 2014, LNBIP 180, pp. 128–135, 2014.

2 Rank Ordering Methods

Different elicitation formalisms have been proposed by which the decision-maker can express preferences. Such a formalism is sometimes based on scoring points, as in point allocation (PA) or direct rating (DR) methods. In PA, the decision-maker is given a point sum, e.g. 100, to distribute among the criteria. Sometimes, it is pictured as putty with the total mass of 100 that is divided and put on the criteria. The more mass, the larger weight on a criterion. In PA, there are consequently $N-1$ degrees of freedom (DoF) for N criteria. DR, on the other hand, puts no limit to the number of points to be allocated. The decision-maker allocates as many points as desired to each criterion. The points are subsequently normalized by dividing by the sum of points allocated. Thus, in DR, there are N degrees of freedom for N criteria.

In [4], there is a discussion on weight approximation techniques which brings the suggestions of rank sum (RS) weights and rank reciprocal (RR) weights. They are suggested in the context of maximum discrimination power, and are both alternatives to ratio based weight schemes. The rank sum is based on the idea that the rank order should be reflected directly in the weight. Assume a simplex S_w generated by $w_1 > w_2 > \ldots > w_N$, where $\Sigma w_i = 1$ and $0 \leq w_i$.[2] Assign an ordinal number to each item ranked, starting with the highest ranked item as number 1. Denote the ranking number i among N items to rank. Then the RS weight becomes for all $i = 1,\ldots,N$

$$w_i^{\mathrm{RS}} = \frac{N + 1 - i}{\sum_{j=1}^{N}(N + 1 - j)} = \frac{2(N + 1 - i)}{N(N + 1)}$$

Another idea discussed is rank reciprocal weights. They have a similar origin as the RS weights, but are based on the reciprocals (inverted numbers) of the rank order for each item ranked. These are obtained by assigning an ordinal number to each item ranked, starting with the highest ranked item as number 1. Then denote the ranking number i among N items to rank and the rank reciprocal (RR) weight becomes

$$w_i^{\mathrm{RR}} = \frac{1/i}{\sum_{j=1}^{N}\frac{1}{j}}$$

A decade later, [5] suggested a weight method based on vertices of the simplex of the feasible weight space. To use the rank order, the ROC (rank order centroid) weights are calculated. These are the centroid components of the simplex S_w. The weights then become the centroid (mass point) components of S_w. The ROC weights are then, for the ranking number i among N items to rank, given by

$$w_i^{\mathrm{ROC}} = 1/N \sum_{j=i}^{N}\frac{1}{j}$$

In this way, it resembles RR more than RS but is, particularly for lower dimensions, more extreme than both in the sense of weight distribution, especially the largest and smallest weights.

---

[2] We will, unless otherwise stated, presume that decision problems are modelled as simplexes S_w.

2.1 A Combined Method

Since these weight models in a sense are opposites, it interesting to see how extreme behaviours can be reduced. A natural candidate for this could be a linear combination of RS and RR. Since we have no reasons to assume anything else, we suggest balancing them equally in an additive combination of the Sum and the Reciprocal weight function that we will call the SR weight method:

$$\mathrm{w}_{\mathrm{i}}^{\mathrm{SR}} = \frac{{}^{1}\!/_{i} + \frac{N+1-i}{N}}{\sum_{j=1}^{N}\left({}^{1}\!/_{j} + \frac{N+1-j}{N}\right)}$$

Of course, other combinations of weights would be possible, but the important results of the paper are obtained using SR and comparing it with others weight functions. For another candidate, the actual mix of the proportions between the methods would affect the results in accordance with its proportions. As will be shown below, all results are sensitive to the underlying assumptions regarding the mind-sets of decision-makers. The SR method is a representative of a class of methods able to handle varying assumptions on decision- maker behaviour.

2.2 Geometric Weights

Geometric weights are based on the idea that the rank order should be reflected multiplicatively in the numeric weights. The multiplicative nature of the geometric weight can be motivated by the likewise multiplicative nature of the terms $w_i^X v_i(a)$ that the overall value $V^X(a) = \sum_{i=1}^{m} w_i^X v_i(a)$, that an alternative a is evaluated by, consist of. Assign an ordinal number to each item ranked, starting with the highest ranked item as number 1. Denote the ranking number *i* among *N* items to rank. Then the geometric sum (GS) weight becomes

$$w_i^{\mathrm{GS}}(\mathrm{s}) = \frac{s^{i-1}}{\sum_{j=1}^{N} s^{j-1}} \quad \text{for } 0 < s < 1$$

As usual, a larger weight is assigned to lower ranking numbers. Similar to some other suggested weight methods, GS contains a parameter *s*.

3 Assessing Models for Surrogate Weights

The underlying assumption of most de facto standard simulation studies is that there exist weights in the decision-maker's mind which are inaccessible by any elicitation method. We will continue this tradition when determining the efficacy, in this sense, of some ranking approaches below. The modelling assumptions regarding decision-makers above are then inherent in the generation of decision problem vectors by a random generator. Thus, following an *N*–1 DoF model, a vector is generated in which the components sum to 100%, i.e., a process with *N*–1 degrees of freedom. Following an *N* DoF model, a vector is generated keeping components within [0%, 100%] and subsequently normalising, i.e., a process with N degrees of freedom. Other distributions modelling actual decision-makers would of course be possible, and could be elicited in one way or another. The important observation is that the validation methods are highly dependent of the model of decision-makers and this produces significant effects on the reliability of the validations.

3.1 Simulation Studies

Thus, in the simulations described below it is important to realize which background model we utilise. As stated above, when following an N–1 DoF model, a vector is generated in which the components sum to 100%. This simulation is based on a homogenous N-variate Dirichlet distribution generator. On the other hand, following an N DoF model, a vector is generated without an initial joint restriction, only keeping components within [0%, 100%] yielding a process with N degrees of freedom. Subsequently, they are normalised so that their sum is 100%. We will call the N–1 DoF model type of generator an *N–1-generator* and the N DoF model type an *N-generator*.

3.2 Comparing the Methods

An N–1 DoF model presents a computer simulation consisting of four steps, assuming the problem is modelled as the simplex S_w.

Generation Procedure

1. For an N-dimensional problem, generate a random weight vector with N components. This is called the TRUE weight vector. Determine the order between the weights in the vector. For each method X' $\in$ {ROC,RS,RR,EW}, use the order to generate a weight vector $w^{X'}$.
2. Given M alternatives, generate $M \times N$ random values with value v_{ij} belonging to alternative j under criterion i.
3. Let w_i^X be the weight from weighting method X for criterion i. For each method X $\in$ {TRUE,ROC,RS,RR,EW}, calculate $V_j^X = \sum_i w_i^X v_{ij}$. Each method produces a preferred alternative, i.e. the one with the highest V_j^X.
4. For each method X' $\in$ {ROC,RS,RR,EW}, assess whether X' yielded the same decision (i.e. the same preferred alternative) as TRUE. If so, record a hit.

It should be noted that most simulation studies to date arrive at the same conclusions regarding ROC, RS, and RR. As we have emphasised above, this is not surprising since different simulations using the same assumptions on degrees of freedom and definitions of weighting methods should (modulo programming errors) yield the same results. A study by [6], though, came up with a different result where RS performed better than ROC with RR in third place. Their paper also suggests a new surrogate weight, ROD, which generates almost identical weights to RS and, thus, performs almost identically. For our purposes, we will consider the latter two equal and will not discuss ROD separately. The random weight distribution in most other simulations (in step 1 of the generation procedure above) is generated by an *N*–1 procedure, thus generating a vector with *N*–1 DoF. Roberts and Goodwin, however, employ a different distribution generating function where a fixed number, say 100, is given to the most important criterion and the others are uniformly generated as U[0,100]. As explained above, this *N*-generator is not the same as *N*–1-generators based on a Dirichlet distribution and thus, their simulation study instead yields the result that RS outperforms ROC with RR in third place. Given an *N*-generator, RS outperforms ROC and RR with EW far behind. ROC is slightly better than RR. While yielding a different "best" weighting method, this result is consistent with the other study results considering it is merely a consequence of choice of DoF in the simulator generator.

Our simulations were carried out with a varying number of criteria and alternatives. There were four numbers of criteria $N = \{3, 6, 9, 12\}$ and five numbers of alternatives $M = \{3, 6, 9, 12, 15\}$ creating a total of 20 simulation scenarios. Each scenario was run 10 times, each time with 10,000 trials, yielding a total of 2,000,000 decision situations generated. For this simulation, an *N*-variate joint Dirichlet distribution was employed to generate the random weight vectors for the *N*–1 DoF simulations and a standard round-robin normalised random weight generator for the *N* DoF simulations. Unscaled value vectors were generated uniformly, and no significant differences were observed with other value distributions. The results of the simulations are shown in the tables below, where we show a subset of the results with chosen pairs (*N*,*M*). The tables show the winner frequency for the six methods ROC, RR, RS, GS, SR, and EW (equal weights) utilising the simulation methods *N*–1 DoF, *N* DoF and a 50% combination of *N*–1 DoF and *N* DoF. All hit ratios in all tables are given in per cent and are mean values of the 10 scenario runs.[3] In Table 1, using an *N*–1-generator, it can be seen that ROC outperforms the others, when looking at the winner, but with GS and SR close behind. RR is better than RS behind the others. In Table 2, the frequencies have changed according to expectation since we employ a model with *N* degrees of freedom. Now RS outperforms all others. SR and GS are close behind while ROC and RR are far behind. In Table 3, the *N* and *N*–1 DoF models are combined with equal emphasis on both. Now, we can see that in total RS, SR, and GS generally perform the best.

Table 1. Using an *N*–1-generator

***N*–1 DoF**	**Winner**	**ROC**	**RS**	**RR**	**GS**	**SR**
3 criteria	3 alternatives	90.2	88.2	89.5	90.0	89.3
3 criteria	15 alternatives	79.1	76.6	76.5	78.2	76.9
6 criteria	6 alternatives	84.8	79.9	82.7	83.9	83.1
6 criteria	12 alternatives	81.3	75.6	78.2	80.0	78.9
9 criteria	9 alternatives	83.5	75.6	79.5	82.0	81.2
12 criteria	6 alternatives	86.4	77.8	80.8	84.8	84.1
12 criteria	12 alternatives	83.4	72.9	76.8	81.4	80.2

[3] The standard deviations between sets of 10 runs were around 0.1-0.3 percent.

Table 2. A model with *N* degrees of freedom

N DoF	Winner	ROC	RS	RR	GS	SR
3 criteria	3 alternatives	87.3	89.3	88.3	88.6	89.1
3 criteria	15 alternatives	77.9	81.1	79.1	80.1	80.6
6 criteria	6 alternatives	80.1	87.3	78.1	84.3	85.1
6 criteria	12 alternatives	76.4	84.3	74.3	81.0	82.0
9 criteria	9 alternatives	76.3	87.2	69.8	82.2	83.0
12 criteria	6 alternatives	77.5	90.3	67.8	84.5	84.6
12 criteria	12 alternatives	73.4	87.6	63.1	80.8	81.7

Table 3. *N* and *N*–1 DoF models are combined

Combined	Winner	ROC	RS	RR	GS	SR
3 criteria	3 alternatives	88.8	88.8	88.9	89.3	89.2
3 criteria	15 alternatives	78.5	78.9	77.8	79.2	78.8
6 criteria	6 alternatives	82.5	83.6	80.4	84.1	84.1
6 criteria	12 alternatives	78.9	80.0	76.3	80.5	80.5
9 criteria	9 alternatives	79.9	81.4	74.7	82.1	82.1
12 criteria	6 alternatives	82.0	84.1	74.3	84.7	84.4
12 criteria	12 alternatives	78.4	80.3	70.0	81.1	81.0

3.3 Introducing Noise

In the above simulations, rankings are induced from the "true" weights. But this assumes that the decision-maker is perfect in converting its belief into orderings, i.e. that the elicitation is perfect. This assumption can at least partly be taken account of by slightly altering the generated "true" weights before the order is generated. For instance, a 10% noise in this sense means that after a generation of a "true" weight vector in step 1 in the generation procedure, the weights are multiplied by a uniformly distributed random factor between 0.9 and 1.1 for the generation of the ranking order (not for the "true" test). By this approach, the size of the change also depends on the true weights. Attributes which have a large true weight will be changed more than attributes which have a small true weight. This in turn will introduce more errors in the important attributes. The generated order in a way simulates that the decision-maker can be uncertain regarding the weight ordering. A measure of robustness can then be that the less affected the method is by this disturbance, the more robust it is.

Table 4. Introducing noise

Combined	Noise	ROC	RS	RR	GS	SR
9 criteria and	0%	79.9	81.4	74.7	82.1	82.1
9 alternatives	10%	79.0	80.7	73.9	81.6	81.5
	20%	78.2	79.8	73.0	80.4	80.3
	30%	76.9	79.0	72.5	79.0	78.8

From Table 4 it can be seen that the behaviour of the respective methods are similar and the hit percentage naturally decreases when the amount of noise increases. Nevertheless, all five methods are quite robust in this sense, even at 30% noise level.

3.4 Discarding Unnatural Decision Situations

It can be argued that the vectors generated by the simulations do not always constitute natural decision problems. For instance, the simulator could generate a weight vector with one component as high as 0.95 and the others correspondingly low. Likewise, the simulator could generate a problem with a weight as low as 0.0001 and such a criterion would probably not be considered at all in real life. Therefore, a filter was designed to discard weight vectors deemed unnatural. Below, we can see the effect of cut-off filters on the simulation results. We used two filters. The weak filter discarded all generated “true” vectors with a component larger than $0.7 + 0.3/N$ or smaller than $0.05/N$. The strong filter discarded all generated “true” vectors with a component larger than $0.6 + 0.25/N$ or smaller than $0.1/N$. If a vector was discarded, a new vector was generated assuring that the total number of vectors remained constant in each simulation. Table 5, Table 6 and Table 7 show the results from applying the weak and strong cut-off filters to three selected decision simulations.

Table 5. Filter with 3 criteria and 3 alternatives

Combined	Cut-off	ROC	RS	RR	GS	SR
3 criteria and	None	88.8	88.8	88.9	89.3	89.2
3 alternatives	Weak	88.5	89.6	89.3	89.5	89.8
	Strong	88.3	90.3	89.4	89.7	90.2

Table 6. Filter with 6 criteria and 12 alternatives

Combined	Cut-off	ROC	RS	RR	GS	SR
6 criteria and	None	78.9	80.0	76.3	80.5	80.5
12 alternatives	Weak	78.8	80.9	76.7	81.3	81.6
	Strong	78.6	82.3	76.8	81.7	82.4

Table 7. Filter with 9 criteria and 9 alternatives

Combined	Cut-off	ROC	RS	RR	GS	SR
9 criteria and	None	79.9	81.4	74.7	82.1	82.1
9 alternatives	Weak	79.9	82.8	75.1	83.0	82.7
	Strong	79.8	83.5	75.4	83.5	83.3

From the tables, it can be seen that most methods improve when faced only with “reasonable” decision situations, the improvement being between 1% and 2%. SR and RS improved the most, with GS and RR less so. The exception is ROC which does not improve at all, rather the hit rate diminishes slightly as extreme decision vectors are cut off.

4 Conclusion

The aim of this study has been to find robust multi-criteria weights that would be able to cover a broad set of decision situations, but at the same time have a reasonably simple semantic regarding how they are generated. To summarise the analysis, we look at the average hit rate in per cent over all the pairs (N,M) that we have reported in the tables above. From the table, it is clear that, considering performance averages, GS and SR are the best candidates when it comes to finding the winning alternative, closely followed by RS. The other surrogate weighs are not in contention. For example, the ROC method relies too heavily on the assumption of decision-makers having an internal decision process with N–1 degrees of freedom for a decision problem with N criteria. Further, the three methods RS, GS, and SR all handle both noise and "unnatural" decision situations equally well. In conclusion, to be robust a rank ordering method should fare well under the varying assumptions. We have above discussed various aspects of performance and it can be seen that the GS and SR methods are the most efficient and robust surrogate weighs that both perform very good on average and are stable under varying assumptions on the behaviour of the decision-maker. Of the two, GS performs a little bit better but is more complex since it requires a parameter to be selected. As simplicity could be regarded an additional sign of robustness, we conclude that GS and SR are equally robust and better choices for surrogate weight functions than the other candidates in the paper.

Acknowledgments. This research was funded by the Swedish Research Council FORMAS, project number 2011-3313-20412-31, as well as by Strategic funds from the Swedish government within ICT – The Next Generation.

References

1. Barron, F., Barrett, B.: Decision Quality Using Ranked Attribute Weights. Management Science 42(11), 1515–1523 (1996)
2. Danielson, M.: Generalized Evaluation in Decision Analysis. European Journal of Operational Research 162(2), 442–449 (2005)
3. Danielson, M., Ekenberg, L.: A Framework for Analysing Decisions under Risk. European Journal of Operational Research 104(3), 474–484 (1998)
4. Stillwell, W., Seaver, D., Edwards, W.: A Comparison of Weight Approximation Techniques in Multiattribute Utility Decision Making. Organizational Behavior and Human Performance 28(1), 62–77 (1981)
5. Barron, F.H.: Selecting a Best Multiattribute Alternative with Partial Information About Attribute Weights. Acta Psychologica 80(1-3), 91–103 (1992)
6. Roberts, R., Goodwin, P.: Weight Approximations in Multi–attribute Decision Models. J. of Multi–Criteria Decision Analysis 11, 291–303 (2002)