

ELABORATION OF A RANKING OF HOTELS

The aim of this project is to develop, through some transparent evaluation models, a ranking of luxury hotels in six European countries. The ranking obtained will be compared to a ranking of luxury hotels given by the website www.booking.com. The data, coming from <https://www.kaggle.com>, are available on <http://www.lamsade.dauphine.fr/~mayag/teaching.html> (see Luxury Hotels Luxe.xlsx file).

For each hotel, we have the following information:

- * **Name:** the name of the luxury hotel;
- * **Total of negative words** (*criterion 1*): the total number of negative words in all the comments given by the clients of this hotel. (*criterion to be minimized*);
- * **Total of positive words** (*criterion 2*): the total number of positive words in all the comments given by the clients of this hotel. (*criterion to be maximized*);
- * **Average given by Reviewers** (*criterion 3*): the average score (/10) given by the clients of this hotels in their comments. (*criterion to be maximized*);
- * **Total of Reviews** (*criterion 4*): the total number of comments given by the clients of this hotel. (*criterion to be maximized*);
- * **Booking's note:** A global score (/10) given the website www.booking.com to this hotel.

The elaboration of a ranking of hotels can be viewed as an elaboration of a MultiCriteria Decision Aid (MCDA) model. We suggest to implement the functions below as generic as possible (for instance, an Excel file containing a set of n criteria and a set of m alternatives).

1 A ranking from a simple weighted sum model

1. Build the function `normalizedperformancematrix1` and `normalizedperformancematrix2` returning an excel file (or csv file) containing the normalized performance matrix of the problem, by using respectively the normalization formula (1) and (2):

$$\begin{cases} u_i(h_i) = \frac{h_i - L_i}{U_i - L_i} & \text{if } i \text{ is a criterion to be maximized (criteria 2, 3 and 4 in our example)} \\ u_i(h_i) = \frac{h_i - U_i}{L_i - U_i} & \text{if } i \text{ is a criterion to be minimized (criterion 1 in our example)} \end{cases} \quad (1)$$

$$\begin{cases} u_i(h_i) = \frac{h_i}{U_i} & \text{if } i \text{ is a criterion to be maximized (criteria 2, 3 and 4 in our example)} \\ u_i(h_i) = 1 - \frac{h_i}{U_i} & \text{if } i \text{ is a criterion to be minimized (criterion 1 in our example)} \end{cases} \quad (2)$$

2. Build a function `normalizedperformancematrix1` which returns, in an Excel (or csv) file, a ranking of luxury hotels by using a weighted sum model. The two formulas (1) and (2) have to be used.

Indication: Weights can be added manually in the Excel file containing the data of the luxury hotels.

3. Test your decision model by using the following weights and compare the results obtained with the ranking provided (see column Booking score) by www.booking.com (for instance, you can use the kendall rank correlation ¹ between the two rankings):

$$\ast (w_1, w_2, w_3, w_4) = (1/4; 1/4; 1/4; 1/4)$$

$$\ast (w_1, w_2, w_3, w_4) = (0.4; 0.2; 0.2; 0.2)$$

$$\ast (w_1, w_2, w_3, w_4) = (0.1; 0.3; 0.4; 0.3)$$

¹https://en.wikipedia.org/wiki/Kendall_rank_correlation_coefficient

2 A ranking from a simple sorting (ordered classification) model

In this section, we sort the luxury hotels in three ordered categories (“Excellent”, “Very Good” and “Good”) by using a simple sorting MCDA method called MR-Sort (for more detail about this MCDA rule, see the Appendix below).

1. Build the functions `PessimisticmajoritySorting` and `OptimisticmajoritySorting`, respectively based on the Pessimistic and Optimistic version of MR-sort rule, which return an Excel (or csv) file containing the classified luxury hotels in the previous three categories.

Indication: Weights, limiting profiles and threshold can be added manually in the Excel file containing the data of the luxury hotels.

2. Test your decision model (in a pessimistic and optimistic way) by using the following parameters:

- ✱ Weights: $(w_1, w_2, w_3, w_4) = (0.4; 0.2; 0.2; 0.2)$
- ✱ Threshold: $\lambda = 0.6$;
- ✱ Limiting profile between “Excellent” and “Very Good”: $(500; 5000; 9; 500)$;
- ✱ Limiting profile between “Very Good” and “Good”: $(1000; 2500; 8; 300)$;

A ELECTRE TRI methods

A.1 Elaboration of the outranking relation $\mathcal{S}_\lambda$

Let A be a set of alternatives evaluated on n real-valued criteria $g_i : A \rightarrow \mathbb{R}, i \in N = \{1, \dots, n\}$. We denote by $g_i(a)$ the performance of the alternative a on criterion i . A nonnegative weight w_i is also assigned to each criterion i (w.l.o.g. we suppose $\sum_{i=1}^n w_i = 1$).

We associate with each criterion $i \in N$, a nonnegative preference threshold $p_i \geq 0$. If the value $g_i(a) - g_i(b)$ is positive but less than p_i , it is supposed that this difference is not significant, given the way g_i has been built. Hence, on this criterion, the two alternatives should be considered indifferent.

Using this information, we define on each criterion $i \in N$ the partial concordance index $c_i : A \times A \rightarrow [0, 1]$ as follows:

$$c_i(a, b) = \begin{cases} 1 & \text{if } g_i(b) - g_i(a) \leq p_i \\ 0 & \text{if } g_i(b) - g_i(a) > p_i \end{cases} \quad (3)$$

The valued relations c_i are aggregated to a single concordance index $c : A \times A \rightarrow \mathbb{R}$ by using the following Equation:

$$c(a, b) = \sum_{i=1}^n w_i c_i(a, b) \quad (4)$$

The binary relation on A called outranking relation is defined by:

$$a \mathcal{S}_\lambda b \text{ iff } c(a, b) \geq \lambda \quad (5)$$

where $\lambda \in [0, 1]$ is a cutting level (usually called a threshold and taken above $\frac{1}{2}$).

Interpretation: An alternative $a \in A$ outranks an alternative $b \in A$ if it can be considered at “least as good” as the latter (i.e., a is not worse than b), given the values (performances) of a and b at the n criteria. If a is not worse than b in every criterion, then it is obvious that $a \mathcal{S}_\lambda b$. However, if there are some criteria where a is worse than b , then a may outrank b or not, depending on the relative importance of those criteria and the differences in the evaluations (small differences might be ignored).

From $\mathcal{S}_\lambda$ we derive the following three binary relations:

☞ “Strictly better than” relation:

$$a \mathcal{P}_\lambda b \text{ iff } [a \mathcal{S}_\lambda b \text{ and not}(b \mathcal{S}_\lambda a)] \quad (6)$$

☞ “Indifferent to” relation:

$$a \mathcal{P}_\lambda b \text{ iff } [a \mathcal{S}_\lambda b \text{ and } (b \mathcal{S}_\lambda a)] \quad (7)$$

☞ “Incomparable to” relation:

$$a \mathcal{P}_\lambda b \text{ iff } [\text{not}(a \mathcal{S}_\lambda b) \text{ and not}(b \mathcal{S}_\lambda a)] \quad (8)$$

A.2 ELECTRE TRI (also called ELECTRE TRI B)

Let us consider r ordered categories $C^1, C^2, \dots, C^r$, C^1 is the worst one and C^r is the best one. The category C^k is modeled by using limiting profiles. The lower limiting profile of C^k is π^k . The upper limiting profile of C^k is π^{k+1} . We suppose that the limiting profiles are such that π^{k+1} strictly dominates π^k ². The profile π^1 (respectively π^{r+1}) is taken low (respectively high). It will be convenient to suppose that $\pi^k \in A$, for each $k = 2, 3, \dots, r$, while $\pi^1, \pi^{r+1} \notin A$. With this convention we have

$$\text{For all } a \in A, a \mathcal{P}_\lambda \pi^1 \text{ and } \pi^{r+1} \mathcal{P}_\lambda a. \quad (9)$$

ELECTRE TRI ([9], chap. 6) renamed ELECTRE TRI-B by Almeida-Dias et al. [4] is a MultiCriteria Decision Aid method using limiting profiles. It has two versions called “pessimistic” and “optimistic” in [9]. In [8] the name “pseudo-conjunctive” is used for the “pessimistic” version and “pseudo-disjunctive” for the “optimistic” version. These two versions are defined as follows:

²An alternative a dominates an alternative b , we note $a \Delta b$ iff [for all $i \in N, g_i(a) - g_i(b) \geq 0$]. a strictly dominates b if [$a \Delta b$ and not($b \Delta a$)]

Définition 1 (Pessimistic version: ETRI-B-pc). *Decrease k from $r + 1$ until the first value k such that $a \mathcal{S}_\lambda \pi^k$. Assign alternative a to C^k .*

ETRI-B-pc assigns an alternative a to the unique category C^k such that a is at least as good as to the lower limiting profile of this category and is not at least as good as its upper limiting profile (the relation “at least as good as” being $\mathcal{S}_\lambda$).

Définition 2 (Optimistic version: ETRI-B-pd). *Increase k from 1 until the first value k such that $\pi^k \mathcal{P}_\lambda a$. Assign alternative a to C^{k-1} .*

ETRI-B-pd assigns an alternative a to the category C^k such that the upper limiting profile of this category is better than a and the lower limiting profile of this category is not better than a (the relation “better than” being $\mathcal{P}_\lambda$).

Remarque 1. Roy and Bouyssou ([9],chap.6,pp.393-395) have shown that if $a \in A$ is assigned to the category C^k by the pessimistic version and to the category C^l by the Optimistic version, then $k \leq l$.

A.3 Majority Rule sorting procedure (MR-Sort)

MR-Sort is a simplified version of the ELECTRE TRI sorting model directly inspired by the work of Bouyssou and Marchant [1, 2] who provide an axiomatic characterization of non-compensatory sorting methods. The general principle of MR-Sort (without veto) is to assign alternatives by comparing their performances to those of profiles delimiting the categories. An alternative is assigned to a category “above” a profile if and only if it is at least as good as the profile on a (weighted) majority of criteria.

The condition for an alternative $a \in A$ to be assigned to a category C^k is expressed as follows:

$$\sum_{i: g_i(a) \geq g_i(\pi^{k-1})} w_i \geq \lambda \text{ and } \sum_{i: g_i(a) \geq g_i(\pi^k)} w_i < \lambda \quad (10)$$

The MR-Sort assignment rule described above involves $r \times n + 1$ parameters, i.e., n weights, $(r - 1) \times n$ profiles evaluations and 1 majority threshold.

As demonstrated in [6], the problem of learning the parameters of a MR-Sort model on the basis of assignment examples can be formulated as a mixed integer linear program (MILP) but only instances of modest size can be solved in reasonable computing times. The MILP proposed in [6] contains $m \times (2n + 1)$ binary variables, with n , the number of criteria, and m , the number of alternatives. A problem involving 1000 alternatives, 10 criteria and 5 categories requires 21000 binary variables. For a similar program in [3], it is mentioned that problems with less than 400 binary variables can be solved within 90 minutes.

In [5] a genetic algorithm was proposed to learn the parameters of an ELECTRE TRI model. This algorithm could be transposed for learning the parameters of a MR-Sort model. However, it is well known in [7] that genetic algorithms which take the structure of the problem into account to perform crossovers and mutations give better results. It is not the case of the genetic algorithm proposed in [5] since the authors’ definitions of crossover and mutation operators are standard.

Learning only the weights and the majority threshold of an MR-Sort model on the basis of assignment examples can be done using an ordinary linear program (without binary or integer variables). On the contrary, learning profiles evaluations is not possible by linear programming without binary variables. Taking these observations into account, [10] proposes an algorithm that takes advantage of the ease of learning the weights and the majority threshold by a linear program and adjusts the profiles by means of a dedicated heuristic. This algorithm uses the following components:

1. a heuristic for initializing the profiles;
2. a linear program learning the weights and the majority threshold, given the profiles;
3. a dedicated heuristic adjusting the profiles, given weights and a majority threshold.

References

- [1] D. Bouyssou and T. Marchant. An axiomatic approach to noncompensatory scoring methods in MCDM, I: The case of two categories. *Eur. J. of Operational Research*, 178:217–245, 2007.
- [2] D. Bouyssou and Th. Marchant. An axiomatic approach to noncompensatory scoring methods in MCDM, II: More than two categories. *Eur. J. of Operational Research*, 178:246–276, 2007.

- [3] O. Cailloux, P. Meyer, and V. Mousseau. Eliciting electre tri category limits for a group of decision makers. *European Journal of Operational Research*, 223(1):133–140, 2012.
- [4] J. Almeida Dias, J. Rui Figueira, and B. Roy. Electre tri-c: A multiple criteria sorting method based on characteristic reference actions. *European Journal of Operational Research*, 204(3):565–580, 2010.
- [5] M. Doumpos, Y. Marinakis, M. Marinaki, and C. Zopounidis. An evolutionary approach to construction of outranking models for multicriteria classification: The case of the ELECTRE TRI method. *European Journal of Operational Research*, 199(2):496–505, 2009.
- [6] A. Leroy, V. Mousseau, and M. Pirlot. Learning the parameters of a multiple criteria sorting method. In Ronen I. Brafman, Fred S. Roberts, and Alexis Tsoukiàs, editors, *Algorithmic Decision Theory - Second International Conference, ADT 2011, Piscataway, NJ, USA, October 26-28, 2011. Proceedings*, volume 6992 of *Lecture Notes in Computer Science*, pages 219–233. Springer, 2011.
- [7] M. Pirlot. General local search methods. *European Journal of Operational Research*, 92(3):493–511, 1996.
- [8] B. Roy. Présentation et interprétation de la méthode ELECTRE TRI pour affecter des zones dans des catégories de risque (p. 25). *Document du LAMSADE*, (124), 2002.
- [9] B. Roy and D. Bouyssou. *Aide multicritère à la décision : Méthodes et Cas*. Economica, 1993.
- [10] O. Sobrie, V. Mousseau, and M. Pirlot. Learning a majority rule model from large sets of assignment examples. In Patrice Perny, Marc Pirlot, and Alexis Tsoukiàs, editors, *Algorithmic Decision Theory - Third International Conference, ADT 2013, Bruxelles, Belgium, November 12-14, 2013, Proceedings*, volume 8176 of *Lecture Notes in Computer Science*, pages 336–350. Springer, 2013.