Reliability Engineering and System Safety ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Robustness and sensitivity analysis in multiple criteria decision problems using rule learner techniques

Claudio M. Rocco S.*, Elvis Hernandez

Universidad Central de Venezuela, Caracas, Venezuela

ARTICLE INFO

Keywords:
Monte Carlo simulation
Multiple criteria decision
Rule learners
Uncertainty

ABSTRACT

In many situations, a decision-maker is interested in assessing a set of alternatives characterized simultaneously by multiple criteria (attributes), and defining a ranking able to synthesize the global characteristics of each alternative, for example, from the best to the worst. This is the case of the assessment of several projects through attributes such as cost, profitability, among others. The behavior of each object, for every criterion, is quantified via numerical or categorical “performance values”. Several multiple criteria decision techniques could be used to this aim. However the base rank could be influenced by uncertain factors associated to specific criteria (e.g., the “ratio Benefit/Cost of a project” could be affected by variations in the interest rate) or by decision-maker preferences. In this situation, the decision-maker could be interested knowing what sets of factors are responsible of specific ranking conditions.

This paper describes the input space of a set of factors responsible of a given model behavior specification, based on the use of rule learners able to provide a description through a set of “**If-Then**” rules derived from model samples. These techniques also allow determining the most important factors. An example related to a real decision problem illustrates the proposed approach.

1. Introduction

In many situations, a decision-maker is interested in assessing a set of m objects or alternatives a_i characterized simultaneously by n criteria or attributes, and defining a ranking able to synthesize the global characteristics of each object, e.g., from the best to the worst. This is the case, for example, in the assessment of several engineering projects through attributes such as cost, availability, environmental impact, among others. The behavior of each object, for every criterion, is quantified via performance values PV_{ij} (for each alternative $i=1, .., m$ and for each criterion $j=1, \dots ,n$) which can either be numerical or categorical.

The idea of ranking alternatives is based on one of the four discrete decision-making problems defined as “*Problematique* γ” in [1], that is, ranking the alternatives from the best to the worst ones. Several multicriteria decision techniques (MC) or ranking techniques could be used to this aim [2]. Ranking techniques to generate the desired rank are classified as parametric and non-parametric. The first group, like ELECTRE [1], PROMETHEE [3], TOPSIS [4] to name a few, requires information about decision-maker preferences (e.g., criterion weights), while non-parametric techniques (partial order ranking [5], Hasse diagram technique [6] and Copeland Scores [7]) do not use such information.

In general, the ranking assessment is performed as follows:

1. Define m alternatives and n criteria.
2. Define the multi-indicator matrix Q, based on each PV_{ij} (for each alternative $i=1, .., m$ and for each criterion $j=1, .., n$).
3. Select a ranking technique.
4. Produce a rank of objects according to the selected technique.

However, no matter which MC technique is selected, the ranking derived using crisp PV (defined as the Base Rank (BR)), could be influenced by uncertain factors associated to specific criteria (for example, the criterion “Cost/Benefit ratio of a project” could be affected by variations in the interest rate) or by decision-maker preferences (e.g., criterion weights).

If these uncertain factors are modeled as a probability distribution function then the rank of each alternative could be considered as a random variable. Several authors [8–11] have analyzed this problem: how the uncertainty in the PV (the input) is propagated or affects the object ranks (the output)?

Recently, Rocco and Tarantola [12] presented two approaches that extend previous works in two directions:

1. Ranking assessment: based on Monte Carlo simulation, the approach allows answering several questions regarding ranking robustness. For example, under uncertainty: What is the probability that the base rank position is maintained? Which is the

* Corresponding author at: Apartado Postal 47937, Los Chaguaramos 1041 A, Caracas, Venezuela. Tel.: +58 412 2528346.
E-mail address: croccoucv@gmail.com (C.M. Rocco S.).

http://dx.doi.org/10.1016/j.ress.2014.04.022
0951-8320/

rank position with the highest probability? What are the possible rank positions and their corresponding probability?

2. Sensitivity analysis: this approach, based on global sensitivity analysis techniques [13], allows evaluating the importance of uncertain factors.

This type of analysis, from input to output, can provide to the decision-maker a sharper picture of the effects of the uncertainty in the final ranking that MC techniques provide. Therefore, the decision-maker can have a better perspective of how stable his/her final decision is and often needs to know which factors determine specific output behavior (output specifications). For example, what are the values associated to each criterion that make a particular project be ranked as the best project?

Procedures to cope with such problems are termed as Factor Mapping setting, "in which specific points/portions of the model output realizations, or even the entire domain, are mapped backwards onto the space of the input factors" [14]. Note that the solution space could be a non-convex and/or sparse space [15–17].

Several approaches have been proposed in the literature to produce such mapping like Monte Carlo Filtering [13,14], Regional Sensitivity Analysis [18], Generalized Likelihood Uncertainty Estimation [19] and Tree-Structured Density Estimation [20].

Other approaches, based on optimization instead of mapping from the output into the input space, have been suggested in [15] or recently in [16,17]. These approaches are able to extract the maximum volume hyperbox of the solution space, where factor variations are assigned independently. The solution space is represented through intervals [x_1_inf, x_1_sup], [x_2_inf, x_2_sup], …, and [x_l_inf, x_l_sup], where x_j is the jth factor and l is the number of uncertain factors considered. Fig. 1 illustrates the approach in the case of two factors x_1 and x_2. The area delimited by dashed lines defines the feasible zone. The rectangle (solid lines) represents the box with maximum area.

The approach proposed in [15] requires an analytical model f (x_1, x_2, .., x_l), while in [16,17] the model is considered as a black-box. In both approaches the hyperbox could be centered at a predefined feasible point or freely centered across the feasible zone. The widths of the final intervals that define the solution space could be considered as a sensitivity index.

This paper proposes an approach based on the use of machine learning classification techniques [21] able to provide a description of the solution space, based on a set of "**If** (*premise*) **then** (*consequence*)" rules derived from model samples (i.e., could be used for analytical or black-box models) where (*premise*) is a condition (or the logical product of several conditions) related to a specific factor or variable whereas (*consequence*) gives a class assignment. For example, for a given project B, the structure of the hypothetical rule

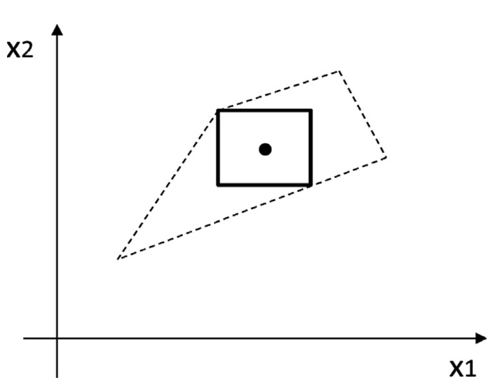

Fig. 1. Maximum area box (solid lines) in the feasible zone defined by dashed lines.

If (Cost/Benefit_Project_B > 8 AND Employment_Project B > 120) **then**
(Rank_project_B = 1)
explains when project B is ranked as the first project.

Each rule extracted represents a specific hyperbox of the solution space. This allows to model non-convex solution spaces. Additionally, some rule generation techniques are able to extract the most important factor, can detect non important factors or can provide a numerical sensitivity index.

The rest of the paper is organized as follows: Section 2 describes the problem to be analyzed and proposes a solution based on machine learning classification techniques. Section 3 presents an overview of rule generations concepts and examples of solutions through several algorithms. Section 4 describes a case study. Finally, Section 5 shows the conclusions and future work.

2. The problem

2.1. Problem statement

Let $A=\{a_1, a_2, \ldots, a_m\}$ be a set of m projects, $G=\{g_1, g_2, \ldots, g_n\}$ the set of n criteria considered in the evaluation with their respective direction of improvement (for example: a high value is better). Each $a_i \in A$ is defined by a set of n values that represents the evaluation of each criterion for the project a_i. In general, each criterion is considered as a defined mathematical function (e.g., availability). In this paper PV_{ij} means the assessment of project i under criterion j.

Let $\mathrm{RI}=\{\mathrm{RI}_1, \mathrm{RI}_2, \ldots, \mathrm{RI}_n\}$ be a set of values that model DM's preferences (or weights) over the selected criteria with $\sum_{j=1}^{n} RI_n = 1, \ \ RI_n \geq 0$.

Let $F()$ be a particular ranking technique: Given PV and RI, F(PV, RI) is able to produce the ranking of a set of projects under study, i.e., $R=[r_1, r_2, \ldots, r_n]^{\mathrm{T}}$, where r_k is the rank position of project k. Although a particular technique is represented as a function $F()$, it does not mean that $F()$ has an analytical definition and it is considered as a "black-box" function. For example, the well known ranking methods PROMETHEE [3] (the technique selected in Section 4 to illustrate the proposed approach) defines a multi-step procedure for

a) normalizing PV;
b) performing all pair-wise comparisons and distance computations among projects;
c) assessing the differences between projects using "generalized criterion" expressions;
d) computing the preference-weighted aggregation and the positive and negative outranking flows for the each project;
e) determining the overall "quality" of each project using the net flows and
f) producing the final ranking R.

Depending on how the preference function is modeled, the PROMETHEE methods may need the definition of additional parameters other than the weights of attributes.

Let R^0 be the base rank obtained when no uncertainty is considered. For example, if $m=4$, $R^0=[3,1,4,2]^{\mathrm{T}}$ means that project 1 occupies the third position, project 2 is the best ranked, and so on. Of course, a reverse ranking order could be used to define the best project.

Suppose that all of the performance values PV and RI are considered as inputs whose uncertainties are modeled as random variables properly characterized through known probability distribution functions (pdf). That means that the R is now a random