Information Fusion 26 (2015) 84–95

Contents lists available at ScienceDirect

Information Fusion

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

A paired-comparison approach for fusing preference orderings from rank-ordered agents

Fiorenzo Franceschini *, Domenico Maisano, Luca Mastrogiacomo

Politecnico di Torino, DIGEP (Department of Management and Production Engineering), Corso Duca degli Abruzzi 24, 10129 Torino, Italy

ARTICLE INFO

Article history:
Received 16 September 2014
Received in revised form 23 January 2015
Accepted 28 January 2015
Available online 7 February 2015

Keywords:
Decision making
Multiple rank-ordered agents
Preference ordering
Ordinal semi-democratic
Paired comparison

ABSTRACT

The problem of aggregating multi-agent preference orderings has received considerable attention in many fields of research, such as *multi-criteria decision aiding* and *social choice theory*; nevertheless, the case in which the agents' importance is expressed in the form of a rank-ordering, instead of a set of weights, has not been much debated. The aim of this article is to present a novel algorithm – denominated as "Ordered Paired-Comparisons Algorithm" (OPCA), which addresses this decision-making problem in a relatively simple and practical way. The OPCA is organized into three main phases: (i) turning multi-agent preference orderings into sets of paired comparisons, (ii) synthesizing the paired-comparison sets, and (iii) constructing a fused (or consensus) ordering. Particularly interesting is phase two, which introduces a new aggregation process based on a priority sequence, obtained from the agents' importance rank-ordering. A detailed description of the new algorithm is supported by practical examples.

1. Introduction

A general decision-making problem is that of aggregating multi-agent preference orderings of different alternatives into a single fused ordering. Assume that M decision-making agents[1] (D_1 to D_M) formulate preference orderings among n alternatives of interest (a, b, c, d, etc.). The objective is to aggregate the M agents' orderings into a single fused ordering, which should reflect them as much as possible; for this reason, the fused ordering can also be defined as *consensus* or *compromise* ordering [1,2]. This aggregation should also take into account the agents' importance, which is not necessarily equal for all of them.

This decision-making problem is very diffused in a variety of real-life contexts, ranging from *multi-criteria decision aiding* [3] to *social choice theory* [4,5]. Some of the reasons for this diffusion are that: (i) preference orderings are probably the most intuitive and effective way to represent preference judgments of alternatives [6], and (ii) they do not require a common scale – neither numeric, linguistic or ordinal – to be shared by the interacting agents [7].

The literature includes a variety of algorithms or aggregation techniques, which can be generally divided in two categories [8]: (i) methods in which all agents have the same importance [9–11], and (ii) methods in which agents have recognized attributes and/or privileged positions of power, represented by weights [3,12–14].

Regarding the second category of methods, in some practical contexts weights are not available and/or their definition can be arbitrary and controversial. For example, weights are often imposed according to political strategies; e.g., the scientific committee of a competitive examination for promotion of faculty members may (arbitrarily) decide that scientific publications will account for 40% of the total performance, research projects for 20%, teaching activity for 30%, etc. Although the literature provides several techniques for guiding weight quantification – for example, the AHP procedure [15,16], the method proposed in [17], or that in [18] – they are often neglected in practice, probably because of their complexity.

For these contexts, the problem of weight assignment is partially overcome by expressing the agents' importance in the form of a rank-ordering – such as $D_1 > (D_2 \sim D_3) > \ldots > D_M$ – instead of a set of weights defined on a *cardinal* scale. In fact, the formulation of such a rank-ordering is certainly simpler and more intuitive than that of a set of weights, especially when the agent importance prioritization is uncertain [6].

This paper will focus on this specific problem, which can be denominated as *ordinal semi-democratic*; the adjective *semi-demo-*

* Corresponding author.
E-mail addresses: fiorenzo.franceschini@polito.it (F. Franceschini), domenico.maisano@polito.it (D. Maisano), luca.mastrogiacomo@polito.it (L. Mastrogiacomo).

[1] By a decision-making agent we will consider any of a wide variety of different types of entities. Examples could be human beings, individual criteria in a multi-criteria decision process, software based intelligent agents on the Internet, etc.

http://dx.doi.org/10.1016/j.inffus.2015.01.004
1566-2535/

cratic indicates that agents do not necessarily have the same importance, while *ordinal* indicates that their rank is defined by a crude ordering. This makes the set of the possible solutions relatively wide, since they may range between the two extreme situations of *dictatorship* – in which the resulting fused ordering basically reflects the preference ordering by the most important agent (dictator) – and *democracy* – where the agents' preference orderings are considered as equi-important.

The ordinal semi-democratic decision-making problem is intriguing for two features: (i) the way the preference orderings are compared, and (ii) the way they are synthesized into a fused ordering, which should also reflect the agents' importance rank-ordering. Despite the adaptability to a large number of practical contexts, this specific problem has received little attention in the literature. Yager [7] proposed an algorithm, hereafter abbreviated as YA (which stands for Yager's Algorithm), which addresses the problem in a relatively simple, fast and automatable way. Unfortunately, this algorithm has some limitations: (i) it is applicable to linear preference orderings only, with neither incomparabilities nor omissions of the alternatives [19], (ii) the resulting fused ordering may sometimes not reflect the preference ordering for the majority of agents [20], and (iii) the fused ordering is determined neglecting an important part of the information available [21]. These limitations will be clarified in the next sections.

The objective of this paper is to introduce a new algorithm, denominated as "Ordered Paired-Comparisons Algorithm" (hereafter abbreviated as OPCA), able to overcome the YA's limitations. The main features of this algorithm are that (i) agents' preference orderings are decomposed into sets of paired comparisons of the alternatives, and (ii) the different importance of agents determines a different priority sequence when comparing and synthesizing these sets into a fused ordering.

The remainder of the paper is organized into three sections. Section 2 recalls the YA in detail. Section 3 illustrates the OPCA. The description of both algorithms is supported by practical examples. Section 4 presents a structured comparison of the two algorithms, aimed at highlighting the advantages of the OPCA with respect to the YA. The concluding section summarizes the original contributions of this paper and its practical implications, limitations and suggestions for future research.

2. Basics of the Yager's Algorithm (YA)

In this section we recall the YA. For a more rigorous description, we refer the reader to the original contribution by Yager [7].

The algorithm can be schematized in the following three basic phases, which are described individually in Sections 2.1, 2.2, 2.3:

- construction and reorganization of preference vectors;
- definition of the reading sequence;
- construction of the fused ordering.

2.1. Construction and reorganization of preference vectors

The YA is applicable to (non-strict[2]) *linear* orderings only. The goal of this phase is building preference vectors based on the preference orderings by the agents. For each agent's vector, we place the alternatives as they appear in the ordering, with the most preferred one(s) in the top positions. If at any point $p > 1$ alternatives are tied (i.e., indifferent), we place them in the same element and then place the null set ("Null") in the next $p - 1$ lower positions. For example, when considering three alternatives (a, b and c) with the ordering $(a \sim b) > c$, the resulting vector will conventionally be $[\{a \sim b\}, \text{Null}, \{c\}]^T$. By adopting this convention, the number (n) of elements of a vector will coincide with the number of alternatives of interest.

Considering four fictitious agents (D_1 to D_4), with four relevant orderings of six alternatives (a, b, c, d, e and f), and assuming a certain (linear) rank-ordering between agents (i.e., $D_4 > (D_2 \sim D_3) > D_1$), the resulting preference vectors can be constructed as shown in Table 1. For simplicity, vectors will be denominated as the relevant agents (i.e., D_i). Each vector element can be associated with an indicator (j) depicting the position/level of the element, in the preference vector.

Next, preference vectors are transformed into "reorganized" vectors, conventionally denominated as D_i^*. This transformation consists in (i) sorting the D_i vectors decreasingly with respect to the agents' importance and (ii) aggregating those with indifferent importance (e.g., D_2 and D_3 in the example) into a single vector. This aggregation is performed through a level-by-level union of the vector elements, where alternatives in elements with the same (j-th) position are considered as indifferent. The resulting D_i^* vectors will therefore have a strictly decreasing importance ordering.

Going back to the example in Table 1, the four vectors (D_1 and D_4) are turned into three reorganised vectors (D_1^* to D_3^*, see Table 2). It can be noted that D_2^* – given by the aggregation of two vectors with equal importance (i.e., D_2 and D_3) – contains two occurrences for each alternative. Of course, the total number of "reorganized" vectors will be smaller than or equal to the number (M) of initial preference orderings (3 against 4 in the example presented).

2.2. Definition of the reading sequence

This phase defines a sequence for reading the elements of the D_i^* vectors, according to the following pseudo-code:

1. Initialise the sequence number to $S = 0$.
2. Consider the elements with lowest position, by setting $j = 1$.
3. Consider the most important D_i^* vector, by setting $i = 1$.
4. Set $S = S + 1$.
5. Associate the element of interest with the sequence number S.
6. If i is lower than the total number of D_i^* vectors, then:
7. Set $i = i + 1$.
8. Consider the element with position j, related to the i-th D_i^* vector.
9. Go To Step 4.
10. End If.
11. If $j < n$ (i.e., total number of alternatives), then:
12. Set $j = j + 1$.
13. Go To Step 3.
14. End If.
15. End.

The sequence defines a *bottom-up* level-by-level reading of vector elements. The first elements read are those with lowest position ($j = 1$). When considering elements with the same (j-th) position, priority is given to the vectors from agents of greater importance. After having read all the elements with (j-th) position, we move up to the ($j + 1$)-th position, repeating the reading sequence. Table 2 reports the sequence numbers (S) associated with each element of the reorganized vectors in the example presented.

2.3. Construction of the fused ordering

This third phase is aimed at determining a fused ordering through a gradual selection of the alternatives. The following pseudo-code illustrates the algorithm for constructing the fused ordering:

[2] The adjective "non-strict" means that these orderings allow the relationship of indifference ("∼") between alternatives. For simplicity, the adjective will be omitted hereafter.