UMONS
Université de Mons

CENTRALE
PARIS

DA2PL' 2012

from Multiple Criteria Decision Aid to Preference Learning

Mons

November 15-16, 2012

1837-2012
175 ans
Faculté Polytechnique

Welcome

Marc Pirlot (UMONS) and Vincent Mousseau (ECP) are welcoming you to the first DA2PL Workshop. The aims of this workshop "*from multiple criteria decision Aid to Preference Learning*" is to bring together researchers involved in Preference Modeling and Preference Learning and identify research challenges at the crossroad of both research fields.

It is a great pleasure to provide, during two days, a positive context for scientific exchanges and collaboration : four invited speakers will make a presentation, twelve papers will be presented, and we will have a poster session and a roundtable. We wish to all participants a fruitful workshop, and an exiting and enjoyable time in Mons.

Marc Pirlot and Vincent Mousseau

Aim of the workshop

The need for search engines able to select and rank order the pages most relevant to a user's query has emphasized the issue of learning the user's preferences and interests in an adequate way. That is to say, on the basis of little information on the person who queries the Web, and, in almost no time. Recommender systems also rely on efficient preference learning.

On the other hand, preference modeling has been an auxiliary discipline related to Multicriteria decision aiding for a long time. Methods for eliciting preference models, including learning by examples, are a crucial issue in this field.

It is quite natural to think and to observe in practice that preference modeling and learning are two fields that have things to say to one another. It is the main goal of the present workshop to bring together researchers involved in those disciplines, in order to identify research issues in which cross-fertilization is already at work or can be expected. Communications related to successful usage of explicit preference models in preference learning are especially welcome as well as communications devoted to innovative preference learning methods in MCDA. The programme of the workshop will consist of three or four invited lectures and about 15 selected research communications.

Support

This workshop is organized in the framework of the GDRI (Groupement de Recherche International) "*Algorithmic Decision Theory*", which is recognized and supported by CNRS (France), FNRS (Belgium), FNR (Luxemburg).

The support of Fonds de la Recherche Scientifique (FNRS, Belgium), Faculté Polytechnique UMONS, Ecole Centrale Paris and Belgian Society for Operational Research (SOGESCI-BVWB) is gratefully acknowledged.

Organization

The DA2PL workshop is jointly organized by Marc Pirlot, University of Mons (UMONS), Faculté Polytechnique, Belgium, and Vincent Mousseau, Ecole Centrale Paris (ECP), France

The workshop is one in a series of events organized for commemorating the 175th anniversary of the foundation of the Faculté Polytechnique de Mons

The Faculté was founded in 1837 by A. Devillez and Th. Guibal, two engineers from Ecole Centrale de Paris ! !

It has been the first engineering school in Belgium (under the name "*Ecole des Mines du Hainaut*")

Program committee

- Raymond Bisdorff (University of Luxembourg, Luxembourg),
- Craig Boutillier (University of Toronto, Canada),
- Denis Bouyssou (Paris Dauphine University, France),
- Ronen Brafman (Ben Gurion University, Israel),
- Bernard De Baets (Ghent University, Belgium),
- Yves De Smet (Université libre de Bruxelles, Belgium),
- Luis Dias (University of Coimbra, Portugal),
- Philippe Fortemps (University of Mons, Belgium),
- Patrick Meyer (Telecom Bretagne, France),
- Vincent Mousseau (Ecole Centrale, Paris),
- Patrice Perny (Pierre and Marie Curie University, France),
- Marc Pirlot (University of Mons, Belgium),
- Ahti Salo (Aalto University, Finland),
- Alexis Tsoukias (Paris Dauphine University, France),
- Aida Valls (Universitat Rovira I Virgili, Catalonia, Spain),
- Paolo Viappiani (Aalborg University, Denmark)

Organizing committee

- Valérie Brison, MATHRO, Faculté Polytechnique, Université de Mons
- Olivier Cailloux, Laboratoire de Génie Industriel, Ecole Centrale Paris
- Yves De Smet, CODE-SMG, Ecole Polytechnique, Université libre de Bruxelles
- Philippe Fortemps, MATHRO, Faculté Polytechnique, Université de Mons
- Massimo Gurrieri, MATHRO, Faculté Polytechnique, Université de Mons
- Vincent Mousseau, Laboratoire de Génie Industriel, Ecole Centrale Paris, France
- Wassila Ouerdane, Laboratoire de Génie Industriel, Ecole Centrale Paris
- Marc Pirlot, MATHRO, Faculté Polytechnique, Université de Mons
- Xavier Siebert, MATHRO, Faculté Polytechnique, Université de Mons
- Arnaud Vandaele, MATHRO, Faculté Polytechnique, Université de Mons
- Laurence Wouters, MATHRO, Faculté Polytechnique, Université de Mons

UMONS
Université de Mons

CENTRALE
PARIS

DA2PL'2012

from Multiple Criteria Decision Aid to Preference Learning

PROGRAM

Thursday November 15th, 2012

9h00 Registration

9h15 Welcoming Address

9h30 Session 1

– Invited speaker : "*Preference Learning : an Introduction*", page 1
Eyke Hüllermeier,
Department of Mathematics and Computer Science, Philipps-Universität Marburg, Germany

The topic of "preferences" has recently attracted considerable attention in artificial intelligence in general and machine learning in particular, where the topic of preference learning has emerged as a new, interdisciplinary research field with close connections to related areas such as operations research, social choice and decision theory. Roughly speaking, preference learning is about methods for learning preference models from explicit or implicit preference information, typically used for predicting the preferences of an individual or a group of individuals. Approaches relevant to this area range from learning special types of preference models, such as lexicographic orders, over "learning to rank" for information retrieval to collaborative filtering techniques for recommender systems. The primary goal of this tutorial is to survey the field of preference learning in its current stage of development. The presentation will focus on a systematic overview of different types of preference learning problems, methods and algorithms to tackle these problems, and metrics for evaluating the performance of preference models induced from data.

10h30 Coffee break

11h00 Session 2

– "*A New Rule-based Label Ranking Method*", pages 3-13
M. Gurrieri[1], X. Siebert[1], Ph. Fortemps[1], S. Greco[2] and R. Slowinski[3]
[1] MATHRO, Faculté Polytechnique, UMONS,
[2] University of Catania, Italy,
[3] Poznan University of Technology, Poland
This work focuses on a particular application of preference ranking, wherein the problem is to learn a mapping from instances to rankings over a finite set of labels, i.e. label ranking. Our approach is based on a learning reduction technique and provides such a mapping in the form of logical rules : if [antecedent] then [consequent], where [antecedent] contains a set of conditions, usually connected by a logical conjunction operator (AND) while [consequent] consists in a

ranking (linear order) among labels. The approach presented in this paper mainly comprises four phases : preprocessing, rules generation, classification and ranking generation.

- "*Preference-based clustering of large datasets*", pages 14-20
  A. Olteanu[1] and R. Bisdorff[1]
  [1] Université du Luxembourg
  Clustering has been widely studied in Data Mining literature, where, through different measures related to similarity among objects, potential structures that exist in the data are uncovered. In the field of Multiple Criteria Decision Analysis (MCDA), this topic has received less attention, although the objects in this case, called alternatives, relate to each other through measures of preference, which give the possibility of structuring them in more diverse ways. In this paper we present an approach for clustering sets of alternatives using preferential information from a decision-maker. As clustering is dependent on the relations between the alternatives, clustering large datasets quickly becomes impractical, an issue we try to address by extending our approach accordingly.

- "*Learning the parameters of a multiple criteria sorting method from large sets of assignment examples*", pages 21-31
  O. Sobrie[1,2], V. Mousseau[1] and M. Pirlot[2]
  [1] LGI, Ecole Centrale Paris,
  [2] MATHRO, Faculté Polytechnique, UMONS
  ELECTRE TRI is a sorting method used in multiple criteria decision analysis. It assigns each alternative, described by a performance vector, to a category selected in a set of pre-defined ordered categories. Consecutive categories are separated by a profile. In a simplified version proposed and studied by Bouyssou and Marchant and called MR-Sort, a majority rule is used for assigning the alternatives to categories. Each alternative a is assigned to the lowest category for which a is at least as good as the lower profile delimiting this category for a majority of weighted criteria. In this paper, a new algorithm is proposed for learning the parameters of this model on the basis of assignment examples. In contrast with previous work ([7]), the present algorithm is designed to deal with large learning sets. Experimental results are presented, which assess the algorithm performances with respect to issues like model retrieval, computational efficiency and tolerance for error.

- "*A piecewise linear approximation of PROMETHEE II's net flow scores*", pages 32-39
  S. Eppe[1] and Y. De Smet[1]
  [1] CoDE, Université Libre de Bruxelles
  Promethee II is a prominent outranking method that builds a complete ranking on a set of actions by means of pairwise action comparisons. However, the number of comparisons increases quadratically with the number of actions, leading to computation times that may become prohibitive for large decision problems. Practitioners generally seem to alleviate this issue by down-sizing the problem, a solution that may not always be acceptable though. Therefore, as an alternative, we propose a piecewise linear model that approximates Promethee II's net flow scores without requiring costly pairwise comparisons : our model reduces the computational complexity (with respect to the number of actions) from quadratic to linear, at the cost of some misranked actions. Experimental results on artificial problem instances show a decreasing proportion of those misranked actions as the problem size increases. This observation leads us to provide empirical bounds above which the Promethee II-ranking of an action set is satisfyingly approximated by our piecewise linear model.

13h00 Lunch

14h30 Session 3

- Invited speaker : "*Principled Techniques for Utility-based Preference Elicitation in Conversational Systems*", page 40
  Paolo Viappiani,

  CNRS-LIP6, Université Pierre et Marie Curie, Paris
  Preference elicitation is an important component of many applications, such as decision support systems and recommender systems. It is however a challenging task for a number of reasons. First, elicitation of user preferences is usually expensive (w.r.t. time, cognitive effort, etc.). Second, many decision problems have large outcome or decision spaces. Third, users are inherently "noisy" and inconsistent.
  Adaptive utility elicitation tackles these challenge by representing the system knowledge about the user in form of "beliefs" about the possible utility functions, that are updated following user responses ; elicitation queries can be chosen adaptively given the current belief. In this way, one can often make good (or even optimal) recommendations with sparse knowledge of the user's utility function.
  We analyze the connection between the problem of generating optimal recommendation sets and the problem of generating optimal choice queries, considering both Bayesian and regret-based elicitation. Our results show that, somewhat surprisingly, under very general circumstances, the optimal recommendation set coincides with the optimal query.

15h30 Coffee break

16h00 Session 4

- "*Using Choquet integral in Machine Learning : what can MCDA bring ?*", pages 41-47
  D. Bouyssou[1], M. Couceiro[1], C. Labreuche[2], J.-L. Marichal[3] and B. Mayag[1]
  [1] CNRS-Lamsade, Université Paris Dauphine,
  [2] Thales,
  [3] Université du Luxembourg.
  In this paper we discuss the Choquet integral model in the realm of Preference Learning, and point out advantages of learning simultaneously partial utility functions and capacities rather than sequentially, i.e., first utility functions and then capacities or vice-versa. Moreover, we present possible interpretations of the Choquet integral model in Preference Learning based on Shapley values and interaction indices.

- "*On the expressiveness of the additive value function and the Choquet integral models*", pages 48-56
  P. Meyer[1] and M. Pirlot[2]
  [1] Institut Télécom, Télécom Bretagne,
  [2] MATHRO, Faculté Polytechnique, UMONS
  Recent - and less recent - work has been devoted to learning additive value functions or a Choquet capacity to represent the preference of a decision maker on a set of alternatives described by their performance on the relevant attributes. In this work we compare the ability of related models to represent rankings of such alternatives. Our experiments are designed as follows. We generate a number of alternatives by drawing at random a vector of evaluations for each of them. We then draw a random order on these alternatives and we examine whether this order is representable by a simple weighted sum, a Choquet integral with respect to a 2- or 3-additive

capacity, an additive value function in general or a piecewise-linear additive value function with 2 or 3 pieces. We also generate non preferentially independent data in order to test to which extent 2- or 3-additive Choquet integrals allow to represent the given orders. The results explore how representability depends on varying the numbers of alternatives and criteria.

- "*Using set functions for multiple classifiers combination*", pages 57-62
  F. Rico[1], A. Rolland[1],
  [1] Laboratoire ERIC - Université Lumière Lyon
  In machine learning, the multiple classifiers aggregation problems consist in using multiple classifiers to enhance the quality of a single classifier. Simple classifiers as mean or majority rules are already used, but the aggregation methods used in voting theory or multi-criteria decision making should increase the quality of the obtained results. Meanwhile, these methods should lead to better interpretable results for a human decision-maker. We present here the results of a first experiment based on the use of Choquet integral, decisive sets and rough sets based methods on four different datasets.

- "*Preference Learning using the Choquet Integral*", page 63
  E. Hüllermeier[1]
  [1] Department of Mathematics and Computer Science, Philipps-Universität Marburg, Germany
  This talk advocates the (discrete) Choquet integral as a mathematical tool for preference learning. Being widely used as a flexible aggregation operator in fields like multiple criteria decision making, the Choquet integral suggests itself as a natural target for learning preference models. From a machine learning perspective, it can be seen as a generalized linear model that combines monotonicity and flexibility in a mathematically sound and elegant manner. Besides, it exhibits a number of additional features, including suitable means for supporting model interpretation. The learning problem itself essentially comes down to specifying the fuzzy measure in the integral representation on the basis of the preference data given. The talk will specifically address theoretical as well as methodological and algorithmic issues related to this problem. Moreover, applications to concrete preference learning problems such as instance and object ranking will be presented.

Friday November 16th, 2012

9h Session 5

- Invited speaker : "*Ranking Problems, Task Losses and their Surrogates*", page 65
  Krzysztof Dembczynski,
  Laboratory of Intelligent Decision Support Systems, Poznan University of Technology

  From the learning perspective, the goal of the ranking problem is to train a model that is able to order a set of objects according to the preferences of a subject. Depending on the preference structure and training information, one can distinguish several types of ranking problems, like bipartite ranking, label ranking, or a general problem of conditional rankings, to mention a few. To measure the performance in the ranking problems one uses many different evaluation metrics, with the most popular being Pairwise Disagreement (also referred to as rank loss), Discounted Cumulative Gain, Average Precision, and Expected Reciprocal Rank. These measures are usually neither convex nor differentiable, so it is, in general, infeasible to optimize them directly. Therefore they are sometimes referred to as task losses, and in the learning algorithms one rather employs surrogate losses to facilitate the optimization problem. The question, however, arises whether we can design for a given ranking problem a surrogate loss that will provide a near-optimal solution with respect to a given task loss. For simple ranking problems and some task losses the answer is positive, but it seems that in general the answer is rather negative. During the talk we will discuss several results obtained so far, with the emphasis on the bipartite and multilabel ranking problem and the pairwise disagreement loss, in which case very simple surrogate losses lead to the optimal solution.

10h00 Coffee break + Poster session

- "*Preference Learning to Rank : An Experimental Case Study*", M. Abbas, USTHB, Alger, Algerie
- "*From preferences elicitation to values, opinions and verisimilitudes elicitation*", I. Crevits, M. Labour, Université de Valenciennes- pages 66-73
- "*Group Decision Making for selection of an Information System in a Business Context*", T. Pereira, D.B.M.M Fontes, Porto, Portugal - pages 74-82
- "*Ontology-based management of uncertain preferences in user profiles*", J. Borras, A. Valls, A. Moreno, D. Isern, Universitat Rovira i Virgili, Tarragona - pages 83-89
- "*Optimizing on the efficient set. New results*", D. Chaabane, USTHB, Alger, Algerie

11h00 Session 6

- Roundtable : "*From Multiple Criteria Decision Analysis to Preference Learning*"
  Participants : E. Hüllermeier, P. Viapianni, K. Dembczynski

12h00 Lunch

13h30 Session 7

- Invited speaker : “*Learning GAI networks*”, page 90
  Yann Chevaleyre,
  LIPN, Université Paris 13

  Generalized Additive Independence (GAI) models have been widely used to represent utility functions. In this talk, we will address the problem of learning GAI networks from pairwise preferences. First, we will consider the case where the structure of the GAI network is known of bounded from above. We will see how this problem can be reduced to a kernel learning problem. Then, we will investigate the structure learning problem. After presenting the computational of algorithms can be used to solve this problem.

14h30 Coffee break

15h00 Session 8

- “*On measuring and testing the ordinal correlation between valued outranking relations*”, pages 91-100
  R. Bisdorff[1],
  [1] University of Luxembourg
  We generalize Kendall’s rank correlation measure to valued relations. Motivation for this work comes from the need to measure the level of ap- proximation that is required when replacing a given valued outranking with a convenient weak ordering recommendation.

- "*Elicitation of decision parameters for thermal comfort on the trains*", pages 101-107
  L. Mammeri[1,2], D. Bouyssou[1], C. Galais[2], M. Ozturk[1], S. Segretain[2] and C. Talotte[2]
  [1] CNRS-Lamsade, Université Paris-Dauphine,
  [2] SNCF
  We present in this paper a real world application for the elicitation of decision parameters used in the evaluation of thermal comfort in high speed trains. The model representing the thermal comfort is a hierarchical one and we propose to use different aggregation methods for different levels of the model. The methods used are rule-based aggregation, Electre Tri and 2-additive Choquet. We show in this paper the reasons of the choice of such methods and detail the approach used for the elicitation of the parameters of these methods.

- "*Dynamic managing and learning of user preferences in a content-based recommender system*", pages 108-114
  L. Marín[1], A. Moreno[1], D. Isern[1] and A. Valls[1]
  [1] Universitat Rovira i Virgili, Tarragona
  The main objective of the work described in this paper is to design techniques of profile learning to enable a Recommender System to automatic and dynamically adapt preferences stored about the users in order to increase the accuracy of the recommendations. The alternatives (or set of possible solutions to the recommendation problem) are defined by multiple criteria that can be either numerical or categorical. A study of the performance of the whole designed techniques so far is also included.

- "*An algorithm for active learning of lexicographic preferences*", pages 115-122
  F. Delecroix[1], M. Morge[1], J.-Chr. Routier[1]
  [1] Université Lille 1
  At the crossroad of preference learning and multicriteria decision aiding, recent research on preference elicitation provide useful methods for recommendation systems. In this paper, we consider (partial) lexicographic preferences. In this way, we can consider dilemmas and we show that these situations have a minor impact in practical cases. Based on this observation, we propose an algorithm for active learning of preferences. This algorithm solve the dilemmas by suggesting concrete alternatives which must be ranked by the user.

17h00 Closing session

Session 1

Invited speaker : Eyke Hüllermeier

Department of Mathematics and Computer Science, Philipps-Universität Marburg, Germany
"***Preference Learning : an Introduction***",

The topic of "preferences" has recently attracted considerable attention in artificial intelligence in general and machine learning in particular, where the topic of preference learning has emerged as a new, interdisciplinary research field with close connections to related areas such as operations research, social choice and decision theory. Roughly speaking, preference learning is about methods for learning preference models from explicit or implicit preference information, typically used for predicting the preferences of an individual or a group of individuals. Approaches relevant to this area range from learning special types of preference models, such as lexicographic orders, over "learning to rank" for information retrieval to collaborative filtering techniques for recommender systems. The primary goal of this tutorial is to survey the field of preference learning in its current stage of development. The presentation will focus on a systematic overview of different types of preference learning problems, methods and algorithms to tackle these problems, and metrics for evaluating the performance of preference models induced from data.

Session 2

- "*A New Rule-based Label Ranking Method*",
  M. Gurrieri[1], X. Siebert[1], Ph. Fortemps[1], S. Greco[2] and R. Slowinski[3]
  [1] MATHRO, Faculté Polytechnique, UMONS,
  [2] University of Catania, Italy,
  [3] Poznan University of Technology, Poland

- "*Preference-based clustering of large datasets*",
  A. Olteanu[1] and R. Bisdorff[1]
  [1] Université du Luxembourg

- "*Learning the parameters of a multiple criteria sorting method from large sets of assignment examples*",
  O. Sobrie[1,2], V. Mousseau[1] and M. Pirlot[2]
  [1] LGI, Ecole Centrale Paris,
  [2] MATHRO, Faculté Polytechnique, UMONS

- "*A piecewise linear approximation of PROMETHEE II's net flow scores*",
  S. Eppe[1] and Y. De Smet[1]
  [1] CoDE, Université Libre de Bruxelles

Reduction from Label Ranking to Binary Classification

Massimo Gurrieri [a,1], Xavier Siebert[a], Philippe Fortemps[a], Salvatore Greco[b]; Roman Słowiński[c]

[a] UMons, Rue du Houdain 9, 7000 Mons, Belgium
[b] Faculty of Economics, University of Catania, Corso Italia 55, 95129 Catania, Italy
[c] Institute of Computing Science, Poznan University of Technology,
3A Piotrowo Street, 60-965 Poznan, Poland

Abstract. This work focuses on a particular application of preference learning, wherein the problem is to learn a mapping from instances to rankings over a finite set of labels, i.e. label ranking. Our approach is based on a learning reduction technique to reduce label ranking to binary classification. The proposed reduction framework can be used with different binary classification algorithms in order to solve the label ranking problem. In particular, in this paper, we present two variants of this reduction framework, one where Multi-Layer Perceptron is used as binary classifier and another one where the Dominance-based Rough Set Approach is used. In the latter, on the one hand it is possible to deal with possible monotonicity constraints and on the other hand it is possible to provide such a mapping (i.e. a label ranker) in the form of logical rules: if [antecedent] then [consequent], where [antecedent] contains a set of conditions, usually connected by a logical conjunction operator (AND), while [consequent] consists in a ranking (linear order) among labels.

Keywords: Label Ranking, Preference Learning, Decision Rules, Dominance-based Rough Set Approach.

1 Introduction

Preference learning [6] is a relatively new topic that is gaining increasing attention in data mining and related fields [8, 9, 10]. The most challenging aspect of this topic is the possibility of predicting weak or partial orderings of labels, rather than single values which is typical of classification problems. Preference learning problems are typically distinguished in three topics: object ranking, instance ranking and label ranking. **Object ranking** consists in finding a ranking function F whose input is a set X of instances characterized by attributes and whose output is a ranking of this set of instances, in the form of a weak order [6]. Such a ranking is typically obtained by giving a score to each $x \in X$ and by ordering instances with respect to these scores. The training process takes as input either partial rankings or pairwise preferences between instances of X. Such a kind of problem is also commonly studied in the field of Multi-Criteria Decision Aid (e.g. the so-called Thierry's choice problem) [19]. In the context of **instance ranking** [6], the goal is to find a ranking function F whose input is a set X of instances characterized by attributes and whose output is a ranking of this set (again a weak order on X). However, in contrast with object ranking, each instance x is associated with a class among a set of classes $C=\{C_1; C_2; ...; C_k\}$ and this set is furthermore ordered (nominal, quantitative or qualitative scales), therefore: $\{C_1 \succ C_2 \succ ... \succ C_k\}$. The output of such a kind of problem consists in rankings wherein instances labeled with higher classes are preferred to (or precede) instances labeled with lower classes. This problem is similar to the problem of *sorting* in the field of Multi-Criteria Decision Aid (e.g. the contact lenses problem) [19]. The learning scenario discussed in this paper concerns a set of training instances (or examples) which are associated with rankings over a finite set of *labels*, i.e. label ranking [4, 5, 6, 7].

[1] Corresponding author.
E-mail address: massimo.gurrieri@umons.ac.be

This paper is organized as follows. In Section 2, we introduce the label ranking topic and existing approaches as well. In particular, we discuss existing learning reduction techniques. In Section 3, we illustrate our reduction framework to reduce label ranking to binary classification and the general classification and ranking generation phases according to our proposed label ranking method. In Section 4, we illustrate two applications of our reduction framework: an application based on the Variable Consistency Dominance-based Rough Set Approach (VC-DRSA) that provides predictions on rankings in the form of decision rules; and an application based on the Multi-Layer Perceptron Algorithm (MLP). In Section 5, we present experimental results that we conducted with three different configurations of our framework. Finally, we present some conclusions in Section 6.

2 Label Ranking

In label ranking, the main goal is to predict for any instance x, from an instance space X, a preference relation $\succ_x$: $X \rightarrow L$ where L= $\{\lambda_1; \lambda_2; ...; \lambda_k\}$ is a set of labels or alternatives, such that $\lambda_i \succ_x \lambda_j$ means that instance x prefers label λ_i to label λ_j. More specifically, we are interested in the case where $\succ_x$ is a total strict order over L, that is, a ranking of the entire set L. Such ranking $\succ_x$ can be therefore identified with a permutation π_x of $\{1, 2, ..., k\}$ in the permutation space Ω of the index set of L, such that $\pi_x(i) < \pi_x(j)$ means that label λ_i is preferred to label λ_j ($\pi_x(i)$ represents the position of label λ_i in the ranking). A complete ranking (i.e. a linear order) for the set L is therefore given by:

$$\lambda_{\pi_x^{-1}(1)} \succ_x \lambda_{\pi_x^{-1}(2)} \succ_x ... \succ_x \lambda_{\pi_x^{-1}(k)} \tag{2.1}$$

where $\pi_x^{-1}(j), j = 1, 2, ...k$, represents the index of the label that occupies the position j in the ranking. In order to evaluate the accuracy of a model (or label ranker), once the predicted ranking π' for an instance x has been established, it has to be compared to the actual true ranking π associated to the instance x, by means of an accuracy measure defined on Ω, the permutation space over L. As explained in [4], it is possible to associate an instance x to a *probability distribution* $\mathbb{P}(.|x)$ over the set Ω so that $\mathbb{P}(\tau|x)$ is the probability to observe the ranking τ given the instance x. The prediction quality of a label ranker M (as in the setting of classification) is typically measured by means of its *expected loss* on rankings:

$$\mathbb{E}(D(\tau_x, \tau'_x)) = \mathbb{E}(D(\tau, \tau')|x) \tag{2.2}$$

where $D(.,.)$ is a distance function (between permutations in our setting) and τ_x and τ'_x are the true outcome and the prediction made by the model M respectively. Given such a distance metric (i.e. loss function) to be minimized, the best prediction is:

$$\tau^* = \arg\min_{\tau' \in \Omega} \sum_{\tau \in \Omega} \mathbb{P}(\tau|x) D(\tau', \tau). \tag{2.3}$$

Spearman's footrule and Kendall's tau are two well known distances between rankings. In a celebrated result [24], it is showed that Spearman's footrule and Kendall's tau are always within a factor of two from each other. Given two permutations $\tau, \tau' \in \Omega$, the *Spearman's footrule* distance is given by:

$$F(\tau, \tau') = \sum_{i=1}^{k} |\tau_i - \tau'_i| \tag{2.4}$$

and measures the total element-wise displacements between two permutations. The *Kendall's tau* distance is instead given by:

$$K(\tau, \tau') = \#\{(i, j) : i < j | \tau_i > \tau_j \wedge \tau'_i < \tau'_j\} \tag{2.5}$$

and measures the total number of pairwise inversions between two permutations. By performing a linear scaling of $K(\tau, \tau')$ to the interval $[-1, +1]$, it is possible to define the *Kendall's tau* coefficient:

$$\tau_k = \frac{n_c - n_d}{\frac{k(k-1)}{2}} \tag{2.6}$$

where n_c and n_d are the numbers of concordant and discordant pairs of labels, respectively. The sum of squared rank distances is also typically used as a distance metric:

$$S(\tau, \tau') = \sum_{i=1}^{k} (\tau_i - \tau'_i)^2 \tag{2.7}$$

and by performing a normalization in the interval $[-1, +1]$, it is possible to define the *Spearman rank correlation* which is a similarity measure between two permutations (rankings):

$$1 - 6\frac{\sum\limits_{i=1}^{k} (\tau_i - \tau'_i)^2}{k(k^2 - 1)}. \tag{2.8}$$

2.1 Label Ranking Approaches

There are two main groups of approaches to label ranking. On the one hand, we have decomposition (or learning reduction) methods, such as *Constraint Classification* [3] and *Ranking by Pairwise Comparisons* [4], that transform the original label ranking problem into a new binary classification problem. On the other hand, we have direct methods that mainly adapt existing classification algorithms, such as *Decision Trees* and *Instance-based learning* [5] (both being lazy methods), *Boosting algorithms* [11] and *Support Vector Machines* (SVM) [12], to treat the rankings as target objects without any transformation over the data set. There also exists an adaptation [21] of the association rule mining algorithm *APRIORI* for label ranking based on similarity measures between rankings and where the label ranking prediction is given in the form of Label Ranking Association Rules: $A \rightarrow \pi$, where $A \subseteq X$ and $\pi \in \Omega$. The main idea of this method is that the support of a ranking π increases with the observation of similar rankings π_i. In this manner it is possible to assign a weight to each ranking π_i in the training set that represents its contribution to the probability that π may be observed.

2.2 Learning Reduction Techniques

As already mentioned, decomposition methods transform the original label ranking problem into one or several binary classification problems, a process that is called *learning reduction* which is a generalization of approaches to multi-label classification [6, 12]. In the approach *Constraint Classification* [3], a utility linear function $f_i(x) = \mathbf{w}^i \bullet x$ is associated to each label λ_i, $\forall i \in \{1, 2, ..., k\}$ and to an instance $x \in X$ and where $\mathbf{w}^i = (w_1^i, ..., w_l^i)$ is an l-dimensional vector consisting of label coefficients associated with label λ_i. The goal is then to find a linear sorting function:

$$h(x) = argsort_{i=1,2,...,k} f_i(x) \tag{2.9}$$

which returns a permutation of the index set $\{1, 2, ..., k\}$ of labels. A preference of the form $\lambda_i \succ_x \lambda_j$ is accordingly converted into a positive constraint $f_i(x) - f_j(x) > 0$ or, equivalently, into a negative one $f_j(x) - f_i(x) < 0$. In this approach, label ranking can be reduced to binary classification by means of Kesler's construction [8]. Constraints can be related to the sign of the inner product: $< z, \mathbf{W} >$, wherein:

$$\mathbf{W} = (\mathbf{w}^1, ..., \mathbf{w}^k) = (w_1^1, ..., w_l^1; ...; w_1^k, ..., w_l^k) \tag{2.10}$$

is an $(k \times l)$-dimensional vector representing the concatenation of all label coefficients and z is a $(k \times l)$-dimensional vector whose components are defined as follows. If for an instance x, $\lambda_i \succ_x \lambda_j$ holds, the components of vector z with index ranging from $((i-1) \times l) + 1$ to $(i \times l)$ are filled with the components of instance x; components with index ranging from $((j-1) \times l) + 1$ to $(j \times l)$ are filled with the opposite of components of instance x; and the remaining entries are filled with 0's. A further component with 1 is added in order to have a positive classification instance. A negative instance is obtained by considering reversed signs. In such a way, each instance x will generate an expanded set *P(x)* given by the union of positive and negative instances. Finally, the entire set of preference instances X generates an expanded training set:

$$P(X) = \cup_{x \in X} (P(x)) \tag{2.11}$$

that is linearly separable by learning a separating hyperplane with any binary classifier. A very important aspect of this approach is that it takes into account the correlation between labels, since the learning dataset contains the overall information about labels. However, this model is very complex and the preprocessing part is quite cumbersome. Kesler's construction multiplies the dimensionality of the data by k and the number of samples by $k - 1$, where k is the cardinality of the label set. It is clear that direct use for training a classifier is in practice not attractive. In *Ranking by Pairwise Comparison* (RPC) [4], instead, the main idea is to explode the original label ranking problem to several independent binary classification problems and to learn a binary classifier for each pair of labels. More particularly, each preference information of the form $\lambda_i \succ_x \lambda_j$ is considered as a training instance for a learner $M_{i,j}$ which outputs 1 if $\lambda_i \succ_x \lambda_j$ holds, 0 otherwise. If $L = \{\lambda_1; \lambda_2; ...; \lambda_k\}$, the number of learners is at most $k(k-1)/2$. The label ranking associated to a new instance is then obtained by means of a voting strategy, wherein each label is, for example, evaluated by summing scores of all learners and then ordered with respect to this evaluation. However the main drawback of this approach is that it trains independent binary classifiers which cannot take into account relations (i.e. correlation) among labels. Consequently, there could be a loss of preference information.

3 Reduction Framework

3.1 Preprocessing: from Label Ranking to Binary Classification

In this section, a general reduction framework is proposed to reduce label ranking to binary classification. Consequently, the proposed reduction framework can be used with different binary classifiers (two variants will be presented in Section 4). In the preprocessing phase, the original label ranking dataset is converted into a new dataset where the original ranking π_x is split into pairwise preference relations. Though the proposed reduction technique is very similar to the one presented in [4], in the latter each pair of labels is treated separately so that $k(k-1)/2$ independent binary classifiers are trained during the training process. By contrast, in our reduction framework, a single classifier is trained and the overall preference information is treated at once in a single dataset, similarly to the reduction schemes used in [25] or in [26].

The original dataset is a set of instances $T = \{(\mathbf{x}, \pi_x)\}$, where $\mathbf{x}$ and π_x represent, respectively, the feature vector and the corresponding target label ranking associated with the instance x. The feature vector $\mathbf{x}$ is in fact an l-dimensional vector $(q_1, q_2, ..., q_l)$ of attributes (typically numerical values), so that: $(\mathbf{x}, \pi_x) = (q_1, q_2, ..., q_l, \pi_x)$. Each original instance $(\mathbf{x}, \pi_x) = (q_1, q_2, ..., q_l, \pi_x)$ is transformed into a set of new (simpler) instances $\{x_{1,2}, x_{1,3}, ..., x_{i,j}, ...\}$ where:

$$x_{i,j} = (q_1, q_2, ..., q_l, p, d) \tag{3.1}$$

with $i, j \in \{1, 2, ..., k\}, i < j$, $p \in \{(\lambda_1, \lambda_2), ..., (\lambda_{k-1}, \lambda_k)\}$ and $d \in \{-1, +1\}$.

This is obtained by splitting the original set of labels into $k(k-1)/2$ pairs of labels and by considering an additional nominal attribute p, called *relation attribute*, whose possible values are all pairs of labels. A decisional attribute $d \in \{-1; +1\}$ is added to take into account the preference relation between two given labels (λ_i,λ_j), represented by the value of *relation attribute p*, according to the ranking π_x. This decision attribute d says in which manner the pair (λ_i, λ_j) has to be considered during the training process. In other words, if for the instance $(\mathbf{x}, \pi_x)$, we want to treat the pair (λ_i, λ_j), then we set $p = (\lambda_i, \lambda_j)$. Moreover, if λ_i is preferred to λ_j then $d = +1$, otherwise if λ_j is preferred to λ_i then $d = -1$. For example, the instance:

$$(\mathbf{x}, \pi_x) = (-1.5, 2.4, 1.6, \lambda_2 \succ \lambda_1 \succ \lambda_3)$$

generates the following set of simpler instances (i.e., a new instance for each possible pair of labels):

$$x_{1,2} = (-1.5, 2.4, 1.6, (\lambda_1, \lambda_2), -1)$$
$$x_{1,3} = (-1.5, 2.4, 1.6, (\lambda_1, \lambda_3), +1)$$
$$x_{2,3} = (-1.5, 2.4, 1.6, (\lambda_2, \lambda_3), +1).$$

The total number of training instances obtained at the end of the reduction (3.1) is $n\frac{k(k-1)}{2}$, where n is the number of original training instances and k is the number of labels, while the total number of attributes is $l+1$ where l is the number of original attributes. By using this reduction framework (3.1), it is therefore possible to treat pair-wise preference information at once in a single dataset where correlations between labels are taken into account simultaneously.

3.2 Classification Process

The classification of an unknown instance x' (i.e. either a testing instance or a new instance to be classified) can be performed by using some binary classifier capable of estimating conditional probabilities (e.g. multilayer perceptron algorithm):

$$\mathbb{P}(\lambda_i \succ_{x'} \lambda_j) = \mathbb{P}(d = 1 | x'^{a}_{i,j}), \tag{3.2}$$

$$\mathbb{P}(\lambda_j \succ_{x'} \lambda_i) = \mathbb{P}(d = -1 | x'^{a}_{i,j}), \tag{3.3}$$

where $x'^{a}_{i,j}$ is the feature vector of x' that is augmented with the relation attribute $p = (\lambda_i, \lambda_j)$, therefore:

$$x'^{a}_{i,j} = (q_1, q_2, ..., q_l, p). \tag{3.4}$$

By using these conditional probabilities, it is possible to define scores:

$$\mathit{\Gamma}^{+}_{(i,j)} = \mathbb{P}(\lambda_i \succ_{x'} \lambda_j) \tag{3.5}$$

$$\mathit{\Gamma}^{-}_{(i,j)} = \mathbb{P}(\lambda_j \succ_{x'} \lambda_i). \tag{3.6}$$

Scores (3.5) and (3.6) represent the probability that for the instance x' label λ_i is ranked higher (preferred to) than λ_j and the probability that label λ_j is ranked higher (preferred to) than λ_i, respectively.

3.3 Ranking Generation Process

The final step of our approach concerns the generation of a final ranking (i.e. a linear order) among the entire set of labels for the new instance x', based on the preference relation $\succ_{x'}$ learned during the classification process. On the one hand, for each pair of labels (λ_i, λ_j), $i, j \in \{1, 2, ..., k\}, i < j$, a decision $d(x')$ is provided and on the other hand, each pair is also associated with scores (3.5) and (3.6). The preference relation learned for pairs of labels is therefore total, asymmetric, irreflexive but, in general, not transitive (i.e. cycles are likely to happen). In order to overcome cycles, a *Net Flow Score* procedure [22] is used to aggregate pairwise preferences. This procedure allows to obtain a linear order among the entire set of labels since each label λ_i is evaluated by considering the following score:

$$S(i) = \sum_{j \neq i} (\mathit{\Gamma}^{+}_{(i,j)} - \mathit{\Gamma}^{-}_{(i,j)}), \tag{3.7}$$

where $\mathit{\Gamma}^{+}_{(i,j)}$ and $\mathit{\Gamma}^{-}_{(i,j)}$ are given by (3.5) and (3.6). The final ranking τ is therefore obtained by ordering labels according to decreasing values of scores (3.7) (the higher the score, the higher the preference in the ranking): $S(i) > S(i) \Leftrightarrow \tau_i < \tau_j$.

4 Applications of Reduction Framework

In this section, we discuss two variants of our reduction framework: an application based on the Variable Consistency Dominance-based Rough Set Approach (VC-DRSA) that provides predictions on rankings in the form of rules; and a variant based on the Multi-Layer Perceptron Algorithm (MLP).

4.1 VC-DRSA for Label Ranking

This variant is based on the rule induction paradigm, i.e. a variant that can provide predictions on rankings in the form of decision rules: $\mathit{\Phi} \rightarrow \mathit{\Psi}$. The rule learner used in this variant is VC-DRSA [13, 14, 15] which is an extension of the Dominance-based Rough Set Approach (DRSA) [2, 23]. The motivation behind this variant is twofold: on the one hand it is due to the fact that most of available methods, though are very efficient algorithms, they lack transparency and mostly perform like "black boxes", i.e. just like oracles which never clearly show relationships between input and output. From this point of view, it seems natural to provide a label ranker in the form of logical rules. It is well known, in fact, that rules clearly show relationships between the feature vector (the antecedent $\mathit{\Phi}$) and the associated output (the consequent $\mathit{\Psi}$), since both are visible in the rule syntax. As a consequence, rules are very easy to interpret and can provide very rich and complete information, especially in the field of decision aid [16, 19, 20]. However, in this context, other rule learners could also be used within our framework as well. On the other hand, the choice of VC-DRSA as a rule learner is also motivated by the potential requirement that an explicit order in the input space (i.e. value-order of attributes) could be used to monotonically establish an order among labels in the output space (i.e. label ranking with monotonicity constraints). We describe in the following the reduction process proposed in this variant which is sligtly different from the reduction (3.1).

Each original instance $(\mathbf{x}, \pi_x) = (q_1, q_2, ..., q_l, \pi_x)$ is transformed into a set of new (simpler) instances $\{x_{1,2}, x_{1,3}, ..., x_{i,j}, ...\}$ where:

$$x_{i,j} = (q_1^{\geq}, q_1^{\leq}, q_2^{\geq}, q_2^{\leq}, ..., q_l^{\geq}, q_l^{\leq}, p, d) \tag{4.1}$$

with $i, j \in \{1, 2, ..., k\}, i < j$, $p \in \{(\lambda_1, \lambda_2), ..., (\lambda_{k-1}, \lambda_k)\}$ and $d \in \{GT, LT\}$.

The main difference consists in transforming each attribute q_h into a gain and a cost criterion $q_h^{\geq}, q_h^{\leq}, h \in \{1, 2, ..., l\}$, to be maximized and minimized, respectively. This can be justified by the fact that the monotonic relationship between a certain attribute and the preference relation between labels is generally unknown. However, in case such a relation is known, one can set a given attribute to be only a gain (cost) criterion. A decisional gain criterion $d \in \{GT; LT\}$ (i.e., respectively, greater than and less than, where $GT \succ LT$) is finally added to take into account the preference relation between two labels (λ_i,λ_j), according to the ranking π_x. This decision criterion says in which manner the relation attribute $p = (\lambda_i, \lambda_j)$ has to be considered for a given instance x. In other words, if for the instance $(\mathbf{x}, \pi_x)$, λ_i is preferred to λ_j, then $d = GT$, otherwise $d = LT$. For example, the instance:

$$x = (-1.5, 2.4, 1.6, \lambda_2 \succ \lambda_1 \succ \lambda_3)$$

generates the following set of simpler instances (i.e., a new instance for each possible pair of labels):

$$x_{1,2} = (-1.5^{\geq}, -1.5^{\leq}, 2.4^{\geq}, 2.4^{\leq}, 1.6^{\geq}, 1.6^{\leq}, (\lambda_1, \lambda_2), LT)$$
$$x_{1,3} = (-1.5^{\geq}, -1.5^{\leq}, 2.4^{\geq}, 2.4^{\leq}, 1.6^{\geq}, 1.6^{\leq}, (\lambda_1, \lambda_3), GT)$$
$$x_{2,3} = (-1.5^{\geq}, -1.5^{\leq}, 2.4^{\geq}, 2.4^{\leq}, 1.6^{\geq}, 1.6^{\leq}, (\lambda_2, \lambda_3), GT)$$

Thus, the original label ranking problem is transformed into a simpler binary classification problem wherein each training instance is represented by criteria instead of attributes, an additional nominal attribute (*relation attribute*) p and a decisional criterion $d \in (GT, LT)$. The final dataset, as shown in Table 1, contains $n \times [k(k-1)/2]$ training instances while the number of conditional criteria is $2l + 1$, where l is the number of original conditional attributes, n is the number of original instances and k is the number of labels.

	$q_1^{\geq}$	$q_1^{\leq}$	..	..	..	$q_l^{\geq}$	$q_l^{\leq}$	**Relation**	**Decision**
$x_{1,2}$	q_1	q_1	..	..	..	q_l	q_l	(λ_1, λ_2)	d
$x_{1,3}$	q_1	q_1	..	..	..	q_l	q_l	(λ_1, λ_3)	d
...	..	..	..	..	..	..	..	..	...
$x_{k-1,k}$	q_1	q_1	..	..	..	q_l	q_l	$(\lambda_{k-1}, \lambda_k)$	d
...			..	..	..				...
...			..	..	..				...

Table 1. Learning Reduction Scheme

4.1.1 Classification Rules As already mentioned, classification rules are sentences of the form:

$$\Phi \rightarrow \Psi$$

where Φ is called the "antecedent" and Ψ is called the "consequent". Φ is typically composed of conditions on the values of some attributes, while Ψ is generally the class to which an instance satisfying the antecedent should be assigned. More complex "antecedent" and "consequent" forms exist as well. For example, in DRSA the antecedent is a conjunction of elementary conditions concerning one or more criteria (either gain or cost criteria) while the consequent relates to either the upper union of totally ordered classes $Cl_t^{\geq}$ or the downard union of totally ordered classes $Cl_t^{\leq}$. When generating rules, two important measures (among others) are usually taken into account to evaluate the quality of a rule. Such measures are *confidence* and *strength*. Let H be the set of training instances verifying the antecedent of a rule r: $\Phi \rightarrow \Psi$ and K the set of training instances verifying the consequent of the rule. We say that rule r holds in the set X with *confidence* $c = \frac{|H \cap K|}{|H|}$ if c is the percentage of instances in H that verify the consequent Ψ. On the other hand, rule r has *strength* $s = \frac{|H \cap K|}{|X|}$ if s is the percentage of instances in X that verify both the antecedent and consequent. Finally, the number $|H \cap K|$ represents the *support* of rule r. In the field of Rough Set Rule Induction several algorithms have been developed in order to generate rules based on the Rough Set Approach (RSA) [1], such as: AQ, LEM2, MLEM2, DomLEM, VC-DomLEM [13, 16, 17]. Such algorithms are typically based on the scheme of a sequential covering [18] and heuristically generate a minimal set of rules covering instances.

4.1.2 The Training Process: Inferring Rules The second phase of our method consists in the inference of rules on pairs of labels based on the training set obtained in the previous phase. A set of rules R is obtained by using VC-DomLEM whose complexitity is polynomial [2]. Let l be the number of attributes and n the number of instances in the original data set, the complexity of the algorithm is given by $nl(n+1)(l+1)/4$ and therefore, the time complexity of the algorithm is in $O(n^2 l^2)$. This algorithm heuristically searches for rules that satisfy a given threshold value of consistency. The applied heuristic strategy is called sequential covering or separate and conquer. It constructs a rule that covers a subset of training instances, removes the covered instances from the training set and iteratively learns another rule that covers some of the remaining instances, until no uncovered instances remain. VC-DomLEM induces an approximately minimal set of minimal decision rules covering all training instances. Since each training instance is considered with each possible pair of labels, the training process ends up with a set of minimal and non-redundant decision rules R covering each possible pair of labels. This set R is comprised of subsets $R_{(1,2)}, R_{(1,3)}, ..., R_{(k-1,k)}$ (one for each pair of labels), where the generic subset $R_{(i,j)}, i, j \in \{1, 2, ..., k\}, i < j$, contains the set of rules $R_{(i,j,GT)}$ for which the decision d associated with the pair (λ_i, λ_j) is GT and the set of rules $R_{(i,j,LT)}$ for which $d = LT$. It is obviuos that $R_{(i,j)} = R_{(i,j,GT)} \cup R_{(i,j,LT)}$ and $R_{(i,j,GT)} \cap R_{(i,j,LT)} = \emptyset$.

4.1.3 Computational Complexity We discuss here the computational complexity of the rule inference process associated to the proposed method. Firstly, we find the number of training instances that are obtained by using the learning reduction technique (4.1) since the total computational complexity depends on this number as well as on the complexity of the rule learner used for processing these instances and generating rules.

Theorem 1 *The time complexity of the inferring rule algorithm (after the learning reduction* (4.1)*) is* $O(\frac{(n^{\frac{k(k-1)}{2}})(2s+1)(n^{\frac{k(k-1)}{2}}+1)(2l+2)}{4})$ *where n is the* *number of original training instances, k is the number of labels and l is the original number of attributes.*

Proof. By applying the reduction (4.1), each original training instance is split into a set of simpler instances $\{x_{1,2}, x_{1,3}, ..., x_{i,j}, ...\}$ which contains exactly $\frac{k(k-1)}{2}$ new instances (one for each possible pair). As a consequence, the total number of training instances obtained at the end of the reduction (3.1) is $n\frac{k(k-1)}{2}$, where n is the number of original training instances and k is the number of labels. Moreover, the total number of attributes is $2l+1$ where l is the number of original attributes. Since the time complexity of VC-DomLEM is given by $O(\frac{nl(n+1)(l+1)}{4})$ [2], by replacing n and l with $n\frac{k(k-1)}{2}$ and $2l+1$, respectively, the total computational complexity of the inferring rules process is $O(\frac{(n\frac{k(k-1)}{2})(2s+1)(n\frac{k(k-1)}{2}+1)(2l+2)}{4})$. □

4.1.4 The Classification Process

The classification process for an unknown instance x' (i.e. either a testing instance or a new instance to be classified) is performed by means of the set R and aims at providing a decision (either *GT* or *LT*) for each pair of labels (λ_i, λ_j), $i, j \in \{1, 2, ..., k\}, i < j$. The classification is performed by considering the following scheme for every pair $(i, j) \in \{1, 2, ..., k\}, i < j$. Let be $R_{(i,j)}$ the set of rules concerning the pair (λ_i, λ_j) and $R_{(i,j,GT)}, R_{(i,j,LT)} \subseteq R_{(i,j)}$ subsets having $d = GT$ and $d = LT$ as decision for the given pair of labels, respectively. For any rule $r \in R_{(i,j)}$, we define the weight:

$$\omega_r = \frac{S_r}{S_{i,j}} \tag{4.2}$$

where $S_{i,j} = \sum_{r \in R_{i,j}} S_r$ and S_r is the *support* of rule r. By using weight (4.2), we define these two scores:

$$W^+_{(i,j)} = \sum_{r \in R_{(i,j,GT)}} \omega_r \tag{4.3}$$

$$W^-_{(i,j)} = \sum_{r \in R_{(i,j,LT)}} \omega_r. \tag{4.4}$$

These two scores represent the total sum of weights of rules for which $d = GT$ and $d = LT$ hold, respectively, for the pair (λ_i, λ_j). Let define, for the testing instance x':

$$x'^a_{i,j} = (q_1^{\geq}, q_1^{\leq}, q_2^{\geq}, q_2^{\leq}, ..., q_l^{\geq}, q_l^{\leq}, p). \tag{4.5}$$

By testing the antecedent of rule r with $x'^a_{i,j}$, it is possible to define:

$$T_{i,j} = \sum_{r \in R_{i,j}} \omega_r \cdot \delta_r^{x'} \tag{4.6}$$

with $\delta_r^{x'} = \begin{cases} 1 \text{ if } x'^a_{i,j} \text{ supports } r, \\ \\ 0 \text{ otherwise} \end{cases}$. The value $T_{i,j}$ is the sum of weights of rules supported by x' for the corresponding pair of labels. Finally, we define:

$$\alpha^+_{(i,j)} = \sum_{r \in R_{(i,j,GT)}} \omega_r \cdot \delta_r^{x'} \tag{4.7}$$

$$\alpha^-_{(i,j)} = \sum_{r \in R_{(i,j,LT)}} \omega_r \cdot \delta_r^{x'} \tag{4.8}$$

which are, respectively, the total sum of weights of rules, supported by x', for which $d = GT$ and $d = LT$ hold for the pair (λ_i, λ_j). Finally, by normalizing weights (4.7),(4.8) in [0,1], we define these two scores:

$$\mathit{\Gamma}^+_{(i,j)} = \frac{\alpha^+_{(i,j)}}{T_{i,j}} \tag{4.9}$$

$$\mathit{\Gamma}^-_{(i,j)} = \frac{\alpha^-_{(i,j)}}{T_{i,j}} \tag{4.10}$$

For a given unknown instance x' (testing instance) and $\forall(\lambda_i, \lambda_j)$, the classification process provides scores $\mathit{\Gamma}^+_{(i,j)}, \mathit{\Gamma}^-_{(i,j)}$. Scores (4.9), (4.10) verify these properties:

$$\mathit{\Gamma}^+_{(i,j)} = \mathit{\Gamma}^-_{(j,i)} \tag{4.11}$$

$$\mathit{\Gamma}^+_{(i,j)}, \mathit{\Gamma}^-_{(i,j)} \in [0, 1] \tag{4.12}$$

$$\mathit{\Gamma}^+_{(i,j)} + \mathit{\Gamma}^-_{(i,j)} = 1 \tag{4.13}$$

So that scores (4.9) and (4.10) can be considered as the probability that for the instance x' label λ_i is ranked higher (preferred to) than λ_j and the probability that label λ_j is ranked higher (preferred to) than λ_i, respectively. That is:

$$\mathit{\Gamma}^+_{(i,j)} \approx \mathbb{P}(\lambda_i \succ_{x'} \lambda_j) \tag{4.14}$$

$$\mathit{\Gamma}^-_{(i,j)} \approx \mathbb{P}(\lambda_j \succ_{x'} \lambda_i) \tag{4.15}$$

4.1.5 Classification Scheme I: Majority Class

The classification process for a new instance x' consists in finding a decision GT or LT for each possible pair of labels (λ_i, λ_j), $i, j \in \{1, 2, ..., k\}, i < j$, by testing the subset of rules $R_{i,j}$. If x' matches at least one rule from $R_{i,j}$, the final decision for the pair (λ_i, λ_j) is chosen by comparing scores (4.9) and (4.10). The decision $d(x)$ is chosen according to $\max(\mathit{\Gamma}^{+}_{(i,j)}, \mathit{\Gamma}^{-}_{(i,j)})$. In case, for a certain pair of labels (λ_i, λ_j), x' does not match any rule from $R_{i,j}$, the model R is silent w.r.t. this pair and therefore the prediction cannot be provided. Two different strategies are presented in this paper in order to deal with this problem.

In the first strategy, we consider a voting procedure consisting in finding the majority class, either GT or LT, for the pair (λ_i, λ_j). The majority class is determined by using scores (4.3) and (4.4) and a *default* decision for x' is associated with the higher score. The classification of instance x' for each pair of labels (λ_i, λ_j), $i, j \in \{1, 2, ..., k\}, i < j$, can be summarized by the following scheme:

$$\textbf{If } \mathit{\Gamma}^{+}_{(i,j)} = \mathit{\Gamma}^{-}_{(i,j)} = 0 :$$

$$d(x') = \begin{cases} \text{GT if } \; W^{+}_{(i,j)} \;\geq\; W^{-}_{(i,j)}; \\ \\ \text{LT } \; \textit{otherwise} \end{cases}$$

$$\textbf{Else}:$$

$$d(x') = \begin{cases} \text{GT if } \; \mathit{\Gamma}^{+}_{(i,j)} \;\geq\; \mathit{\Gamma}^{-}_{(i,j)}; \\ \\ \text{LT } \; \textit{otherwise} \end{cases}$$

4.1.6 Classification Scheme II: One Nearest Neighbor

The second strategy is based on the One Nearest Neighbor scheme (1NN). If for a new instance x' the prediction (decision) $d(x')$ for a certain pair of labels cannot be provided by the model R, the nearest testing instance $x*$ is selected from the training set (the Euclidean distance is used as the distance metric) and $d(x') = d(x*)$. Moreover, x' is associated with same the scores as $x*$. The classification of instance x' for each pair of labels (λ_i, λ_j), $i, j \in \{1, 2, ..., k\}, i < j$, can be summarized by the following scheme:

$$\textbf{If } \mathit{\Gamma}^{+}_{(i,j)} = \mathit{\Gamma}^{-}_{(i,j)} = 0 :$$

$$d(x') = d(x*),$$

(where $x*$ is the nearest neighbor of x' in the training set)

$$\textbf{Else}:$$

$$d(x') = \begin{cases} \text{GT if } \; \mathit{\Gamma}^{+}_{(i,j)} \;\geq\; \mathit{\Gamma}^{-}_{(i,j)}; \\ \\ \text{LT } \; \textit{otherwise} \end{cases}$$

4.1.7 Ranking Generation Process

The ranking generation process is the same as the one discussed in 3.3. As discussed above, the set of rules R provides, for each pair of labels (λ_i, λ_j), $i, j \in \{1, 2, ..., k\}, i < j$, a decision $d(x')$ and scores (4.9) and (4.10), which can be approximated with conditional probabilities (4.14), (4.15). The final ranking τ is therefore obtained by ordering labels according to decreasing values of scores (3.7) (the higher the score, the higher the preference in the ranking): $S(i) > S(i) \Leftrightarrow \tau_i < \tau_j$.

4.2 Multi-layer Perceptron for Label Ranking

As discussed above, our reduction framework can be used with any binary classifier as long as it can provide conditional probabilities (3.5), (3.6). In particular, in our experiments, we run a configuration of our reduction framework with Multi-layer Perceptron algorithm (MLP) [8, 9] which provides good estimates of conditional probabilities [27, 28, 29]. Since the perceptron training is based on the minimization of the (least square) error, its output can be viewed as estimation of probability, which is approximated by the perceptron as a result of training. Weka machine learning package was used for the implementation of this variant.

5 Experiments and Discussion

This section is devoted to experimental studies that we conducted in order to evaluate the performance of our method in terms of its predictive accuracy. The data sets used in this paper were taken from KEBI Data Repository [2]. Some information about the data sets is provided in Table 2. The evaluation measures used in this study are the *Kendall's*

[2] see http://www.uni-marburg.de/fb12/kebi/research/repository

DATA SETS	#Instances	#Labels	#Attributes
Glass	214	6	9
Iris	150	3	4
Vehicle	846	4	18
Vowel	528	11	10
Wine	178	3	13

Table 2. Summary of the datasets

Kendall's Tau	RBLR	RBLR+	MLPLR	ARLR	RPC	CC	LL	IBLR	LRT
Glass	.882(3)	**.906(1)**	.863(4)	.850(5)	.882(3)	.846(6)	.817(8)	.841(7)	.883(2)
Iris	.956(4)	.961(2)	**.971(1)**	.960(3)	.885(6)	.836(7)	.818(8)	.960(3)	.947(5)
Vehicle	.812(7)	.863(2)	**.870(1)**	.750(8)	.854(5)	.855(4)	.601(9)	.859(3)	.827(6)
Vowel	.776(5)	**.897(1)**	.858(2)	.720(7)	.647(8)	.623(9)	.770(6)	.851(3)	.794(4)
Wine	.883(8)	.901(7)	.931(4)	.910(6)	.921(5)	.933(3)	.942(2)	**.947(1)**	.882(9)
Average Rank	.861(5)	**.905(1)**	.898(2)	.838(6)	.837(7)	.818(8)	.789(9)	.891(3)	.866(4)

Table 3. Comparison of RBLR, RBLR+, MLPLR with state-of-the-art methods (Kendall's Tau)

tau (2.6) and the *Spearman Rank Correlation coefficient* (2.8). The performance of the method was estimated by using a cross validation study (10-fold, 5 repeats). In this section, the performance of our rule-based label ranking method is compared to the performances of constraints classification (**CC**) [3], pairwise comparison(**RPC**) [4], log-linear (**LL**) [11], association rules for label ranking (**ARLR**) [21], instance based learning (**IBLR**) [5] and decision tree for label ranking (**LRT**) [5]. It should be pointed out that we did not run the experiment on the other methods. Our results have been simply compared with published results of the other methods. However, even if results cannot be directly compared, they can provide some indications of the quality of our method in comparison to the state-of-the-art. The experimental results, in terms of (2.6) and (2.8), are discussed hereinafter. In this experiment, we considered three different configurations of our approach. In the basic version (**RBLR**), we generated certain rules by using VC-DomLEM with a consistency level of 0.98 and we used the majority class strategy for classification. This version, though not very efficient in terms of prediction accuracy, is the simplest one. Another configuration of the present method is the One Nearest Neighbor (**RBLR+**) version, where instead of using the majority class strategy for classification, we used the 1NN class strategy. A third configuration (**MLPLR**) was implemented by condidering the Multilayer perceptron (MLP) as binary classifier where scores (4.9) and (4.10) are associated with the distribution probabilities of classes +1 and −1. Results (shown in Table 3, Table 4, Table 5) clearly show that the present method is very competitive to other state-of-the-art methods in terms of prediction accuracy. In particular, **RBLR+** shows better performances with respect to other methods. Apart from performance results, our method has several advantages w.r.t. other methods: the modularity of the architecture, since any binary classifier can be used (as long as probability distributions can be provided) and the simplicity of our reduction framework.

6 Conclusions and Future Work

In this paper we presented a new approach to label ranking, which is based on a learning reduction technique.

The contributions of this paper can be summarized as follows. We developed a general reduction framework to reduce label ranking to binary classification that can be solved by some binary classifier (e.g. rule based learners, multilayer perceptron). In particular, the dataset associated with the label ranking problem is reduced to a single binary classification dataset so that the overall preference relation on labels is treated at once instead of exploding it into several independent binary classifiers. In a specific variant of our approach, we generate a label ranker in the form of a set of logical rules. In this variant, it is also possible to take into account potential monotonicity constraints between the input and the output (i.e. preferences on labels). Compared to other methods, this variant of our approach is more appropriate for real-world applications since it gives clear and directly interpretable results to an end user. By

Kendall's Tau	RBLR+	MLPLR
Glass	.906 ±.006	.863 ±.044
Iris	.961 ±.002	.971 ±.003
Vehicle	.863±.003	.870 ±.031
Vowel	.897 ±.017	.858 ±.018
Wine	.901±.001	.931 ±.043

Table 4. Performance of RBLR+, MLPLR in terms of Kendall's tau (mean and standard deviation)

Spearman's Rank	RBLR+	MLPLR
Glass	.928 ±.005	.890 ±.046
Iris	.971 ±.001	.977±.025
Vehicle	.891 ±.003	.896±.025
Vowel	.945 ±.012	.924±.013
Wine	.919±.011	.948 ±.032

Table 5. Performance of RBLR+, MLPLR in terms of Spearman's rank (mean and standard deviation)

using this kind of model, the user could be invited to analyze rules that are activated for a given query, i.e. some instance profile or a specific preference relation between pairs of labels. The activated rules show which *scenarios* of *cause-effect* relationships match the considered query. For example, suppose that this following rule is obtained: $\textbf{IF}[(q_2 \geq 2.4) \wedge (q_3 \leq 1.9)]\textbf{THEN}(\lambda_2 \succ \lambda_3)$. This rule not only gives a prediction on the preference relation between labels λ_2 and λ_3, but it can also serve to argument the recommendation (in this case the reason why $\lambda_2 \succ \lambda_3$). Moreover, this rule could also be used to select a specific feature vector (e.g. a specific user's profile): which kind of input does verify this preference relation? In other words, by knowing that $\lambda_2 \succ \lambda_3$ holds whenever $[(q_2 \geq 2.4) \wedge (q_3 \leq 1.9)]$, one could activate a query to search a specific group (i.e. a specific target audience) having $[(q_2 \geq 2.4) \wedge (q_3 \leq 1.9)]$. Such a kind of strategy could be useful, for example, in marketing and advertising for reaching target markets. Finally, the approach presented in this paper is very competitive compared to other existing methods in terms of prediction accuracy. There are several directions for future work in order to improve the approach discussed w.r.t. computational complexity and efficiency. On the other hand, other possibilities could be investigated with regard to the generation of final rankings.

References

1. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data, Dordrecht, (1991).

2. Greco, S., Matarazzo, B., Słowiński, R. and Stefanowski, J.: An algorithm for induction of decision rules consistent with dominance principle, in W. Ziarko, Y. Yao (eds.): Rough Sets and Current Trends in Computing, LNAI 2005, Springer, Berlin, pp. 304-313, (2001).

3. Har-Peled, S., Roth, D. and Zimak, D., Constraint classification for multiclass classificatin and ranking in Advances in Neural Information Processing Systems, pp. 785-792, (2002).

4. Hüllermeier, E., Fürnkranz, J., Cheng, W. and Brinker, K.: Label Ranking by learning pairwise preference. Artif. Intell. 172 (16-17), 1897-1916, (2008).

5. Cheng, W., Hühn, J., Hüllermeier, E.: Decision Tree and Instance-Based Learning for Labele Ranking. Proc. ICML-09, International Conference on Machine Learning. Montreal, Canada, (2009).

6. Fürnkranz, J., and Hüllermeier, E., (eds.): Preference Learning, Springer-Verlag, 2010.

7. Gärtner, T., Vembu, S.: Label Ranking Algorithms: A Survey. In Johannes Fürnkranz, Eyke Hüllermeier, editor(s), Preference Learning, Springer-Verlag, (2010).

8. Duda, R. O., Hart, P. E., and Stork D. G.: Pattern Recognition. John Wiley Sons, (2000).

9. Witten, I. H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Published by Morgan Kaufmann, (2011).

10. Tan, PN., Steinbach, M. and Kumar, V., Introduction to Data Mining. Published by Addison Wesley Longman, (2006).

11. Dekel, O., Manning, C. D., and Singer, Y.: Log-linear models for label ranking. In Advances in Neural Information Processing Systems 16, (2003).

12. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In Advances in Neural Information Processing Systems 14, (2001).

13. Błaszczyński, J., Słowiński, R. and Szeląg, M.: Sequential Covering Rule Induction Algorithm for Variable Consistency Rough Set Approaches. Information Sciences, 181, 987-1002, (2011).

14. Błaszczyński, J., Greco, S., Słowiński, R. and Szeląg, M.: Monotonic variable consistency rough set approaches. International Journal of Approximate Reasoning, 50, 979999, (2009).

15. Błaszczyński, J., Greco, S. and Słowiński, R.: Multi-criteria classification - a new scheme for application of dominance-based decision rules. European Journal of Operational Research, 181, 10301044, (2007).

16. Doumpos, M., Zopounidis, C.: Multicriteria Decision Aid Classification Methods. Applied Optimization, Volume 73, 15-38, (2004).

17. Jerzy, W., Grzymala-Busse: Rough Sets and Intelligent Systems Paradigms Lecture Notes in Computer Science, Volume 4585/2007, 12-21, (2007).

18. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, Morgan Kaufmann, San Francisco, CA, (2006).

19. Vincke, Ph.: "L'aide Multicritère à la décision" Editions de l'ULB - Ellipses (1988).

20. Jacquet-Lagreze, E., Siskos, Y.: Preference disaggregation: 20 years of MCDA experience. EJOR 130, 233-245, (2001).

21. Sá, C. et al. Mining Association Rules for Label Ranking. Proceeding PAKDD'11 Proceedings of the 15th Pacific-Asia conference on Advances in knowledge discovery and data mining - Volume Part II, (2011).

22. Bouyssou, D. Ranking methods based on valued preference relations: a characterization of the net flow method, European Journal of Operational Research, 60, 61-68, 1992.

23. Greco, S., Matarazzo, B., Słowiński, R.: Rough sets theory for multicriteria decision analysis, European Journal of Operational Research, vol 129 (2001), pp. 1-47.

24. Diaconis, P., Graham, R. L.: Spearmans footrule as a measure of disarray, Journal of the Royal Statistical Society, Series B (Methodological), 39 (1977). 262-268.

25. Schapire, R.E.; Singer, Y.: Improved boosting algorithms using confidence-rated predictions. Proceedings of the Eleventh Annual Conference on Computational Learning Theory (Madison, WI, 1998), 8091 (electronic), ACM, New York, 1998.

26. Lin, H., Li, L. (2012): Reduction from cost-sensitive ordinal ranking to weighted binary classification. Neural Computation, 24, 13291367.

27. Hung, M.S., Hu, M.Y., Shanker, M.S., Patuwo B.E.: Estimating Posterior Probabilities In Classification Problems With Neural Networks. International Journal of Computational Intelligence and Organizations, 1(1), 49-60, 199.

28. Cybenko, G.: Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems, 2:303314, 1989.

29. Funahashi, K.I.: On the approximate realization of continuous mappings by neural networks. Neural Networks, 2:183192, 1989.

Preference-based clustering of large datasets

Alexandru-Liviu Olteanu [1 2 3] and **Raymond Bisdorff** [1]

Abstract. Clustering has been widely studied in Data Mining literature, where, through different measures related to similarity among objects, potential structures that exist in the data are uncovered. In the field of Multiple Criteria Decision Analysis (MCDA), this topic has received less attention, although the objects in this case, called alternatives, relate to each other through measures of preference, which give the possibility of structuring them in more diverse ways.

In this paper we present an approach for clustering sets of alternatives using preferential information from a decision-maker. As clustering is dependent on the relations between the alternatives, clustering large datasets quickly becomes impractical, an issue we try to address by extending our approach accordingly.

1 Introduction

In the field of Multi-Criteria Decision Aid we can identify three classical types of problems [18]: choice, ranking and sorting. The first consists in constructing a best choice recommendation (ex: selecting a car to buy), the second aims at building an order, partial or weak, on a set of decision alternatives (ex: ordering candidates for a job position from best to worst), while the last type of problem tries to assign the alternatives to a predefined set of classes (ex: placing students into 'good', 'medium' or 'bad' categories).

Clustering is generally defined as an unsupervised process of grouping objects together. Therefore it relies on certain measures, classically related to similarity, and groups the objects together based on the simple logic of placing similar objects in the same cluster and placing those that are dissimilar in different ones. As there is no interaction with a real person during this process, it is also generally desired that the number of groups be found automatically. Clustering has been used in many fields, such as artificial intelligence, information technology, image processing, biology, psychology, marketing and others. Due to this diversity in the fields of application, and different requirements, many clustering approaches have been developed. For a thorough presentation of clustering methods the reader should refer to [10].

In MCDA the alternatives have additional information on them from that in Data Mining, the Decision Makers (DM) preferences. A review of the existing clustering methods in MCDA can be found in [5]. There, the authors first classify the clustering approaches in two: those that don't use the full range of preferential information and those that do. The latter are also split into relational and ordered clustering approaches, where the first type propose relations between clusters, while the latter constructs also an order on these relations.

Among the classical clustering approaches applied to the field of MCDA we mention here the efforts of BISDORFF [2] who actually proceeds to cluster the criteria and not the alternatives. This approach makes use of a similarity based proximity index for comparing the criteria together, and extracts the clusters as kernels in the graph derived from this index which is cut at a median level.

DE SMET and GUZMAN have extended the classical K-MEANS algorithm to the MCDA context in [7], however they don't propose a way to construct the relations between the different clusters. In [6], this work has been extended to propose such relations. In both cases the authors consider that a crisp outranking relation between the alternatives is given.

FIGUERA et. al. also extended the K-MEANS algorithm in a multicriteria framework [9], and a more recent effort on extending this classical algorithm can be found in [1].

NEMERY and DE SMET also proposed a clustering approach that finds a set of ordered clusters in [14]. A more recent work on this topic was done by FERNANDEZ et. al. in [8]. In these approaches, the order between the clusters is complete, however ROCHA et. al. [16] have worked very recently on a method that is able to find sets of partially ordered clusters.

Some clustering methods in MCDA don't use the additional preferential information while others suffer from common drawbacks of clustering methods from Data Mining (number of clusters need to be specified beforehand, falling into locally optimal solutions, etc.). We therefore explore the problem of clustering in MCDA by using the preferential information that is available between alternatives. We present several objectives for clustering in this context, which we consider to be of interest and propose a method to find them. We then consider the issues related to the complexity of this approach and propose an extension in order to deal with large datasets. We would also like to mention that the approach does not require the number of clusters to be given beforehand and looks for the optimal result by making use of a meta-heuristic approach.

The potential applications of this work, as generally with most classical clustering approaches, lie in exploratory analysis. We may therefore imagine using the presented approach to cluster a large set of alternatives and present in the end to the Decision Maker a summary of the entire dataset through a representative for each cluster plus some additional measures supporting this information. This may done in a context where the DMs preferences have been extracted beforehand, in the form of a preference model. However, it is impossible for a real person to consider a large set of alternatives at one time, also finding a sample that represents well the original dataset and is small enough for the DM to consider may be very difficult. Hence, we may apply our approach to extracting a preference model of the DM by starting with some standard values for the parameters of the considered model (or with some general values for them given by the DM), and construct a set of clusters. We may then confront

[1] CSC/ILIAS, University of Luxembourg, Faculty of Science, Technology and Communications, 6 Rue Coudenhove-Kalergi, L-1359, Luxembourg, Luxembourg

[2] Institut Télécom, Télécom Bretagne, UMR CNRS 6285 Lab-STICC, Technopôle Brest Iroise, CS 83818, 29238 Brest Cedex 3, France

[3] Université Européenne de Bretagne, France

the DM with the summary of the clustering results and elicit information based on this summary. In this way we achieve also a level of abstraction from the original data, where the DM may focus on certain alternatives he is more familiar with and neglect the others.

The article is structured in the following way. We first outline the ways in which alternatives can be compared together, and focus on one such approach. Then we present a way to construct such relations between clusters followed by defining the clustering objectives. In Section 3 we outline the method that we propose and its extension, while finally in the last part of this article we validate our clustering algorithm using empirical results obtained from solving a set of artificially generated benchmarks.

2 Defining the problem of clustering

In this section we define formally the problem of clustering in MCDA and the different potential structures it can uncover.

2.1 Comparing alternatives

We consider the set of alternatives $X = \{x, y, z, ...\}$, which are evaluated on a family of criteria $F = \{1, 2, ..., m\}$. The evaluation of alternative x on criterion i is denoted through x_i. We assume without loss of generality that all criteria have ratio measurement scales on the $[0, 1]$ interval, and that the preference direction on all of them is increasing. Many ways of comparing these alternatives exist which use preferential information from a DM. The two main directions are those that construct value functions [11] or outranking relations [17], though from a general perspective, all of them draw the following conclusions between two alternatives: *indifference* (the alternatives are equivalent), *strict preference* (one alternative is better than the other) and *incomparability* (there is insufficient support for any of the previous situations). Note that the last preferential relation can only appear when using outranking relations.

We will use from this point on the bipolar-valued outranking relation S in order to compare the alternatives together. This relation is constructed using several parameters which are used to model the preferences of a DM (significance weights, indifference, preference and veto thresholds) and has attached to it a bipolar-valued characteristic function r. Due to the way in which it is constructed, the value of the characteristic function for the converse of the S can be easily extracted from the value for the S by reversing the sign. For more details on this relation the reader should refer to [3]. Using this characteristic function, the following statements can be made:

$$\begin{aligned} x \mathrel{\mathrm{S}} y &\iff r(x \mathrel{\mathrm{S}} y) > 0 \\ x \mathrel{\not\mathrm{S}} y &\iff r(x \mathrel{\mathrm{S}} y) < 0. \end{aligned} \tag{1}$$

This means that an alternative outranks another if the value of the characteristic function attached to this statement is strictly positive. A similar judgement can be made with respect to the fact that an alternative does not outrank another, while if the value of r is equal to 0 then neither of the statements can be made, and therefore a situation of indetermination occurs.

From this relation only the relations of indifference, denoted with I and that of preference, denoted with P can be constructed. Similar to [19], these relations are constructed as follows, $\forall x, y \in X$:

$$\begin{aligned} x \mathrel{\mathrm{I}} y &\iff x \mathrel{\mathrm{S}} y \; and \; y \mathrel{\mathrm{S}} x \\ x \mathrel{\mathrm{P}} y &\iff x \mathrel{\mathrm{S}} y \; and \; y \mathrel{\not\mathrm{S}} x. \end{aligned} \tag{2}$$

The absence of a relation between two alternatives is attributed to an indetermination.

The I relation is reflexive and symmetric while P is asymmetric.

We may extend the characteristic function r to these relations through:

$$\begin{aligned} r(x \mathrel{\mathrm{I}} y) &= \min\big(r(x \mathrel{\mathrm{S}} y), r(y \mathrel{\mathrm{S}} x)\big) \\ r(x \mathrel{\mathrm{P}} y) &= \min\big(r(x \mathrel{\mathrm{S}} y), -r(y \mathrel{\mathrm{S}} x)\big) \end{aligned} \tag{3}$$

We notice that only the credibility of one of the $x \mathrel{\mathrm{I}} y$, $x \mathrel{\mathrm{P}} y$ and $y \mathrel{\mathrm{P}} x$ statements will be strictly positive at a given time, except for the case when two or more of these credibilities will be equal to 0. Therefore, the way in which the credibility of these relations have been constructed is consistent with their definitions.

2.2 Comparing sets of alternatives

Having shown the way in which we compare alternatives together we define a way of extending this to sets of alternatives.

For any two sets of alternatives $C, D \subseteq X$, we extend the characteristic function for the S relation through:

$$r(C \mathrel{\mathrm{S}} D) := \frac{\sum\limits_{\substack{x \in C, y \in D \\ x \neq y}} r(x \mathrel{\mathrm{S}} y)}{|C| \cdot |D| - |C \cap D|} \tag{4}$$

Following this, we can make the same statements as in the case of pairs alternative, related to the outranking relation S, in Equation (1), but between sets of alternatives. We extend the indifference and preference relations to sets of alternatives as follows:

$$\begin{aligned} C \mathrel{\mathrm{I}} D &\iff C \mathrel{\mathrm{S}} D \text{ and } D \mathrel{\mathrm{S}} C \\ C \mathrel{\mathrm{P}} D &\iff C \mathrel{\mathrm{S}} D \text{ and } D \mathrel{\not\mathrm{S}} C. \end{aligned} \tag{5}$$

In the case where an indetermination occurs with respect to the outranking relations between the two sets, we give precedence to the relation of preference. In addition, both outranking relations may be indeterminate, therefore the choice of the direction of the relations of preference is made randomly. The characteristic functions for these relations between sets of alternatives are:

$$\begin{aligned} r(C \mathrel{\mathrm{I}} D) &= \min\big(r(C \mathrel{\mathrm{S}} D), r(D \mathrel{\mathrm{S}} C)\big) \\ r(C \mathrel{\mathrm{P}} D) &= \min\big(r(C \mathrel{\mathrm{S}} D), -r(D \mathrel{\mathrm{S}} C)\big) \end{aligned} \tag{6}$$

One important property related to the P relation between sets of alternatives, which we may wish to obtain when clustering, is that of transitivity. We first define the characteristic function for detecting if for three disjoint sets of alternatives $C, D, E \subseteq X$ this property holds:

$$Tr(C, D, E) := \max\Big(r(C \mathrel{\mathrm{P}} E), -\min\big(r(C \mathrel{\mathrm{P}} D), r(D \mathrel{\mathrm{P}} E)\big)\Big) \tag{7}$$

Following this, the property of transitivity of the P relation on a partition K of X may be checked if for all triples of sets from K the above property holds. We define the crisp characteristic function of transitivity as follows:

$$Tr(K) := \begin{cases} 1 & \text{, if } \min\limits_{\substack{C,D,E\in K\\ C\neq D\neq E}} Tr(C,D,E) > 0, \\ -1 & \text{, otherwise.} \end{cases} \tag{8}$$

If $Tr(K)$ is 1 then the P relation is transitive on K, whereas if this function is -1 then the relation is not transitive on K.

2.3 Clustering objectives

We will now focus on the different structures that could be uncovered through the process of clustering in MCDA.

Before this we remind several particular types of binary relations:

- identity relation: S is reflexive, symmetric and antisymmetric;
- tournament: S is asymmetric and complete;
- strict total order: S is asymmetric, complete and transitive;

In Data Mining, clustering is very generally defined as the process of grouping objects that are similar and separating those that are dissimilar. However, in MCDA we compare alternatives with respect to a decision-makers preferences. The complementary notions of similarity and dissimilarity are no longer the object of clustering and are replaced by those of indifference, preference or in some cases incomparability. From these notions, the indifference is the only one that can be used to bring alternatives together, while the rest are used to set them apart.

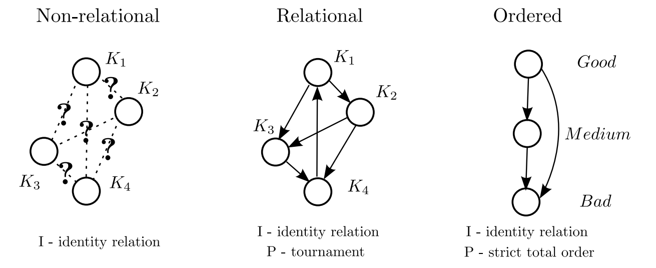


Figure 1. Classification of clustering objectives;

Clustering in MCDA is therefore the process of grouping alternatives that are indifferent and separating those that are preferred or incomparable. Additionally we may try to place certain relations between clusters and enforce properties on them. The result is that clustering in MCDA may yield different types of results. We present a classification of the different clustering objectives in Figure 1.

Non-relational clustering

We begin with the simplest clustering objective, the one where we look only at the level of indifference between alternatives.

Non-relational clustering is the process that groups alternatives that are indifferent and separates those that are not.

At this point we search for groups of alternatives which are indifferent to each other with a higher degree of confidence than with the alternatives from other groups. Notice that we are not concerned with the relations between the clusters, therefore we call this non-relational clustering.

A non-relational clustering result can be described with respect to the binary relation I as a partition K of X where I is an identity relation on K.

We may model a fitness function that measures how well the result is supported by the valued indifference relation through:

$$f_{nr}(K) := \frac{2}{|X|(|X|-1)}\Big(\frac{|C|(|C|-1)}{2}\sum_{C\in K} r(C\,\mathrm{I}\,C) - |C||D|\sum_{C\neq D\in K} r(C\,\mathrm{I}\,D)\Big). \tag{9}$$

The first factor is used to bring this measure to a $[-1, 1]$ interval, while the other two scaling factors account for the number of relations that the I relation summarises. The first term adds positively the support of the indifference relations inside each cluster while the second adds those between different clusters negatively. This function needs to be maximized in order to find the ideal clustering result based on our definition of non-relational clustering.

The practical applications of such a process would be to determine the different groups of alternatives from which, if a decision-maker would select such a group, he would have very little drawbacks in taking any of the alternatives inside to substitute the entire group with it. This clustering objective can thus have a purpose of summarising and compressing the original dataset.

Relational clustering

We may restrict the definition of non-relational clustering by trying to find a result where the relation between the clusters is more structured and has certain properties.

Since the relation of indifference has a role of bringing the alternatives together, and so we try to maximize its support inside a cluster, we may not use it also to define the relation between clusters. Given the case that $r(C\,\mathrm{I}\,D) > 0$ for any two sets $C, D \subseteq X$, merging them would yield a set that will contain a positive support of I, therefore a good cluster. As a result, only P will be used to define the relation between clusters.

Relational clustering is the process that groups alternatives that are indifferent and separates those that are preferred to others in one direction or the other.

A relational clustering result can be described with respect to the binary relations I and P as a partition K of X where I is a an identity relation on K and P is a tournament on K.

We model the fitness function for this clustering objective through:

$$f_r(K) := \frac{2}{|X|(|X|-1)}\Big(\frac{|C|(|C|-1)}{2}\sum_{C\in K} r(C\,\mathrm{I}\,C) - |C||D|\sum_{C\neq D\in K} \max\big(r(C\,\mathrm{P}\,D), r(D\,\mathrm{P}\,C)\big)\Big). \tag{10}$$

We acknowledge the fact that this two clustering objective may not hold many practical applications except, as that of non-relational clustering, to provide a description of the dataset.

Ordered clustering

A particular group of relational clustering objectives look for the relation P between clusters to be transitive, therefore ordering the clusters from best to worst.

Ordered clustering is the process that groups alternatives that are indifferent and separates those that are preferred to others in one direction or the other in such a way that the clusters are ordered.

This clustering result is translated into the partition K of X on which I is an identity relation and P is a strict total order.

The fitness of this clustering objective is defined as:

$$f_o(K) := \min\big(Tr(K), f_r(K)\big). \tag{11}$$

This fitness is the same as that from relational clustering, with the exception that it is brought to 0 if the relation between clusters is not transitive.

This objective provides results that may be compared to those from ranking problems in MCDA, as the alternatives are ordered from best to worst, allowing for ties to occur. It be viewed as a ranking procedure where the result of placing an alternative at a certain level in the ranking is supported by the relations between it and the others.

3 Solving the problem of clustering

As most clustering problems are difficult to solve in an exact manner, enumerating all partitions of the data in order to find the optimal one with respect to the fitness functions defined so far is impractical. For this reason we will present in the following a meta-heuristic approach for solving this problem.

3.1 Brief overview of the clustering approach

We propose an extension of our previous work on clustering on similarities [4] to clustering on the richer information given by the MCDA context. We mention here also the extension to preferentially ordered clustering in [13]. We will not detail the approach in too great detail as it is very similar to our most recent work. In addition, any method that groups alternatives that are indifferent and separates those that are not can be employed at this stage, as the main focus of this paper is placed on the extension to clustering large datasets.

The method is split in two parts:

1. building an initial partition based on the relations of indifference between alternatives;
2. refining this partition by taking in account also the relations of preference and incomparability between the alternatives.

This step aims at building an initial partition where inside each cluster alternatives are predominantly indifferent to each other, and between different clusters there are few or no indifference relations. The problem can thus be defined as finding dense regions in the graph constructed from the indifference relations denoted with $G(X, I)$. The approach we have proposed first looks at finding cores in this graph, which are represented as alternatives that are all indifferent to each other, i.e. cliques, and that are consistently indifferent and not indifferent to the same alternatives from the dataset. Based on these characteristics the step that follows the selection of the cores builds the initial partition by adding the remaining alternatives to the core to which they are linked by the largest number of indifference relations. This step is greedy in nature and is motivated by the refinement step that follows.

The second step of the clustering approach consists in a meta-heuristic, related to simulated annealing, which moves one alternative from one cluster to another in order to bring the result closer to the optimal value of the fitness function for a given clustering objective.

When faced with clustering large datasets, taking into account all the alternatives and the entirety of the relations between them quickly becomes impractical. As a result we propose to proceed to cluster only a subset of the original data. From this result we then extract certain information which can then be used by a much simpler procedure that will split the entire dataset.

We have selected the two most pertinent clustering objectives to extend in such a way: non-relational and strict complete order clustering.

3.2 Non-relational clustering of large datasets

We remind that non-relational clustering aims at grouping alternatives that are indifferent and separating those that are not. As a result we may characterise a group of indifferent alternatives by a single one.

Considering a partition K of X, which contains k sets of alternatives, we define a **central profile** for each set of alternatives $K_l, \forall l \in 1..k$, and denote it through c_l. This central profile will be characterised through a high level of indifference to the alternatives of the cluster to which it is assigned.

We may extract this central profile in two ways, either by selecting an existing one from $K_l, \forall l \in 1..k$, or by constructing it.

We define the fitness of a central profile through:

$$f_c(c_l, K_l) := \sum_{x \in K_l} r(x \,\mathrm{I}\, c_l) \tag{12}$$

Selecting an alternative from K_l to become a central profile for this set can be summarised as:

$$c_l := \arg\max_{x \in K_l} f_c(x, K_l) \tag{13}$$

If several alternatives from K_l equally have the highest value of f_c then one of them is selected at random.

The second approach consisting in the construction of a fictive alternative will potentially yield a central profile with a higher value of this function and in the worst case will not improve this value becoming identical to one of the alternatives from the set K_l.

We present in the following the general outline of the meta-heuristic used to infer a central profile. We have chosen to adapt simulated annealing [12] to our problem, though we could apply any single-solution based meta-heuristic by using the fitness function defined before and the set of operations that we can make when constructing the central profile, which will be presented further.

The simulated annealing meta-heuristic is based on the process of heating and then slowly cooling a substance in order to obtain a strong crystalline structure. It can be described by the series of steps in Algorithm 1.

The algorithm starts with an initial solution and an initial temperature variable. As this temperature variable decreases towards its minimum value, a series of operations are made on the current solution. A neighbour is randomly generated and if it improves the fitness function from that of the old solution then it replaces it. The algorithm also allows for non-improving neighbours to be selected, with a probability proportional to the decrease in fitness and to the temperature variable. Therefore, in the beginning of the algorithm, non-improving solutions have a higher probability of being selected, therefore we tend to explore the solution space, while towards the end the algorithm converges to a final solution. For each temperature level a fixed number of steps may be performed.

Algorithm 1 Simulated Annealing meta-heuristic;

```
$s \leftarrow$ InitialSolution()
$T \leftarrow T_{max}$
while $T > T_{min}$ do
    while not EquilibriumCondition() do
        $s^{'} \leftarrow$ GenerateRandomNeighbour($s$)
        if $f(s^{'}) - f(s) \geq 0$ then
            $s \leftarrow s^{'}$
        else if $random(0,1) > e^{\frac{f(s)-f(s^{'})}{T}}$ then
            $s \leftarrow s^{'}$
        end if
    end while
    $T \leftarrow$ UpdateTemperature($T$)
end while
return $s$
```

In out case, s will be c_l, the central profile of cluster K_l. In the initialization step we select the alternative from K_l which has the highest value of the fitness defined in Equation (15). Each neighbour is then generated by changing the value of c_l on one criterion with the smallest amount, either positively or negatively, which would yield a change in the way in which it compares to the alternatives in K_l. Therefore, there are a number of $2 \cdot |F|$ possible operations on the current solution. As the number of criteria usually used in any decision problem is generally small, we proceed to generate all neighbours and then select one as the next solution.

In keeping with the outranking philosophy, each of the mentioned changes of the current solution is evaluated through a voting procedure. Each alternative in K_l will act as a voter, placing a vote to increase or decrease the evaluation of c_l on each criterion $i \in F$. Of course, each voter may sustain from this process if no change to c_l is required from the perspective of that particular alternative. In addition each voter that needs for the evaluation of c_l on criterion i to change also gives the amount with which this change should be made in order to make a difference in the way it compares to the profile. The result of this vote will give the direction in which the evaluation should be changed and also the smallest amount needed. The fitness of this operation will be equal to the majority margin that decided the result of the vote.

In order to explore also non-improving solutions, a neighbour which changes c_l in contrast to the result of the vote is also created, but it will have a fitness equal to 0. When there is a tie between the votes, both neighbours will be created in this way and will have a fitness equal to 0. In case no vote has been cast for a particular change of c_l on i, a new evaluation is generated at random in order to continue exploring the solution space.

We present below the rules on which each alternative $x \in K_l$ votes on the increase or decrease of the evaluation of c_l on criterion i.

- $c_{li} - x_i \geqslant v_i$: → decrease c_{li} to $x_i + v_i - \epsilon$
- $c_{li} - x_i \geqslant p_i$: → decrease c_{li} to $x_i + p_i - \epsilon$
- $c_{li} - x_i > q_i$: → decrease c_{li} to $x_i + q_i$
- $c_{li} - x_i < -q_i$: → increase c_{li} to $x_i - q_i$
- $c_{li} - x_i \leqslant -p_i$: → increase c_{li} to $x_i - p_i + \epsilon$
- $c_{li} - x_i \leqslant -v_i$: → increase c_{li} to $x_i - v_i + \epsilon$

If we take for example the first case, the evaluation of c_l on i is much greater than that of x, by an amount larger then the veto threshold v_i. In this case this evaluation needs to be lowered. However, as we are only looking at the pair of alternatives x and c_l, we proceed in a prudent manner and propose to decrease it by the smallest amount that will change the result of this comparison. For this purpose we have also place the strictly positive constant $\epsilon \ll 1$.

After the initial generation of the neighbours, the initial temperature of the annealing process is computed followed by the main loop. In here a neighbour is selected through different mechanisms such as roulette wheel selection, or stochastic universal sampling to name a few. Afterwards, the new individual is evaluated with respect to the globally best solution, a new set of neighbours is generated and the temperature of the system is updated. This temperature has the role of giving initially higher probabilities to non-improving neighbours, probability which gets lower as the system cools. This is translated in an affinity to explore the solution space at early stages followed by a convergence towards a final solution towards the end.

With the exception of the creation of the neighbours, all the other steps of this approach are based on standards in the simulated annealing literature.

Being able to characterise any set of alternatives through a central profile, to which the alternatives in the set have a high level of indifference, we may proceed to proposing a method of clustering large datasets.

In order to reduce the complexity of the problem we propose to cluster into K using the presented method a sample of the original dataset. Different strategies for drawing the sample can be used, however, in out implementation we have chosen a simple random selection approach.

After clustering the sample, we construct a central profile for each cluster and the proceed to assigning the alternatives in the original dataset to the clusters defined by these profiles following the rule:

$$x \in K_m : \; m = \underset{l \in 1..k}{\arg\max}\, r(x \, \mathrm{I} \, c_l), \forall x \in X. \qquad (14)$$

Therefore, each alternative will be placed in the cluster to whose profile it is most indifferent. As a result the complexity of clustering the initial large dataset becomes linear.

3.3 Ordered clustering of large datasets

Ordered clustering aims at grouping alternatives that are indifferent and creating an order between these groups.

A parallel can be easily made between the ordered clustering problem and that of sorting. When sorting, the alternatives are grouped into categories based on the relations between them and several alternatives that are external to the original dataset. Usually these alternatives are given by the decision-maker, or they are inferred from assignment examples [15]. These alternatives are called profiles, and can be either central or delimiting. Sorting will only consider the relations between the alternatives in the original dataset and these profiles, whereas clustering takes into account only the relations between the alternatives.

We will consider in what follows the ElectreTri methodology [20], which follows two assignment rules: pessimistic and optimistic.

The pessimistic assignment rule places an alternative in the lowest category whose lower delimiting profile it outranks.

The optimistic assignment rule places an alternative in the highest category whose upper delimiting profiles outranks it.

For a given set of k clusters K, ordered from best to worst, we define the delimiting profiles $l_l, \forall l \in 1..k-1$. There is no need for an upper profile for the best cluster or a lower profile for the worst cluster. l_1 will represent the lower profile of the best cluster, and the upper one of the second best cluster.

We define the fitness of a delimiting profile, considering either the pessimistic or the optimistic assignment rules through:

$$f_l^{pes}(l_l, K) := \sum_{i=1}^{k} \sum_{x \in K_i} r(x \,\mathrm{S}\, l_l) - \sum_{i=k+1}^{k-1} \sum_{x \in K_i} r(x \,\mathrm{S}\, l_l)$$

$$f_l^{opt}(l_l, K) := \sum_{i=k+1}^{k-1} \sum_{x \in K_i} r(l_l \,\mathrm{S}\, x) - \sum_{i=1}^{k} \sum_{x \in K_i} r(l_l \,\mathrm{S}\, x) \quad (15)$$

Using these fitness functions we employ the same meta-heuristic approach to construct these profiles as in the case of non-relational clustering. The only difference is that we will consider all the profiles at once, looping from on to another as opposed to the case of non-relational clustering, where we considered only one at a time.

We detail here the way in which the neighbours are generated, which differs from the previous approach. Each alternative will again vote to increase or decrease the value on a particular criterion $i \in F$ of a delimiting profile l_l considering in this case all the alternatives in X. Depending on the assignment procedures described before, each alternative will vote based on the need to ensure that an outranking relation is validated or not between them and a particular profile.

Considering that alternative $x \in X$ should outrank profile l_l, then the following situations may occur on a particular criterion $i \in F$:

- $l_{li} - x_i \geqslant v_i$: → decrease l_{li} to $x_i + v_i - \epsilon$
- $l_{li} - x_i \geqslant p_i$: → decrease l_{li} to $x_i + p_i - \epsilon$
- $l_{li} - x_i > q_i$: → decrease l_{li} to $x_i + q_i$

In case alternative x should not outrank l_l we have the following rules:

- $l_{li} - x_i < -q_i$: → increase l_{li} to $x_i - q_i$
- $l_{li} - x_i \leqslant -p_i$: → increase l_{li} to $x_i - p_i + \epsilon$
- $l_{li} - x_i \leqslant -v_i$: → increase l_{li} to $x_i - v_i + \epsilon$

Similar rules can be derived for the optimistic assignment procedure.

We may again apply a similar approach as for non-relational clustering, where we draw a sample from the original dataset, cluster it using our approach to find an ordered clustering result, followed by the inference of the delimiting profiles and the sorting of the original dataset using an ELECTRETRI approach. In this case too, the complexity of clustering the original dataset is brought to a linear one.

4 Empirical results

We will present in this section several empirical results generated by running our approach on several artificially constructed benchmarks.

We have created three groups of 100 benchmarks containing 1000, 5000 and 10000 alternatives respectively, that are defined on 10 criteria. We have used qualitative measurement scales for all criteria on an interval from 0 to 1.

When constructing each problem instance, each alternative may fall into one of three categories: good, medium or bad. This corresponds to the values of the alternative on all criteria being generated based on normal distributions with either higher, or lower central points. These distributions do overlap, therefore there exists a chance that alternatives from the good category may be similar to certain alternatives from the medium category. The same is true for the medium and bad categories.

On these benchmarks we then proceeded to drawing a sample of 50 alternatives, which we then clustered using our approach. We did so for both non-relational and ordered clustering approaches described in this paper. Following this step we then inferred the central profiles in the case of the non-relational clustering approach and the delimiting profiles in the case of ordered clustering, following both the pessimistic and the optimistic assignment procedures. In the end, the entire datasets were grouped following the principles described in the previous section.

Both steps of clustering the sample and of inferring the profiles were given one minute of computational time, and we have repeated the simulations 100 times over each benchmark.

We highlight below the results on all the benchmarks using the Rand Index, which show how well the clusters found by our approach match the original classes.

Table 1. Average Rand Index (standard deviation in brackets);

Clustering objective	Profile type	Instance size 1000	5000	10000
non-rel.	central	0.91 (0.05)	0.92 (0.05)	0.91 (0.05)
ordered	pessimistic	0.87 (0.06)	0.87 (0.06)	0.87 (0.06)
ordered	optimistic	0.72 (0.09)	0.72 (0.09)	0.71 (0.09)

We find that the proposed methods work with the same performance across all sizes of the datasets. This is expected, as the way in which the datasets are constructed is exactly the same for all sizes. However, we find that the ordered clustering approaches have lower performances than the non-relational clustering one, which we attribute to the assignment procedures which are not completely consistent with the manner in which the ordered clustering results were initially constructed. This issue should be explored further.

5 Conclusions

In this paper we have outlined the problem of clustering in the field of MCDA, and three possible structures that can be extracted from a set of alternatives. We have briefly presented a method for solving this problem and its extension to clustering large datasets, through the use of central or delimiting profiles. These profiles both have the property of summarizing the original set of alternatives. They can also be used in conjunction with simpler classifiers to cluster large datasets, when these profiles have been constructed by clustering a small sample of the original dataset.

The results look promising, though we would like at this stage to test this approach on benchmarks where the alternatives are structured in more diverse ways then those presented in this paper. This also brings the issue of using a good sampling technique, which we need to explore further. Finally we envision a new definition of the delimiting profiles which is more in accordance to the construction of the ordered clustering results, potentially constructing two delimiting profiles, upper and lower, for each cluster independently of the others.

REFERENCES

[1] R. Baroudi and N.B. Safia, 'Towards multicriteria analysis: A new clustering approach', in *Machine and Web Intelligence (ICMWI), 2010 International Conference on*, pp. 126 –131, (2010).

[2] R. Bisdorff, 'Electre-like clustering from a pairwise fuzzy proximity index', *European Journal of Operational Research*, **138**(2), 320–331, (2002).

[3] R. Bisdorff, 'On polarizing outranking relations with large performance differences', *Journal of Multi-Criteria Decision Analysis*, 1–20, (in press).

[4] R. Bisdorff, P. Meyer, and A.-L. Olteanu, 'A clustering approach using weighted similarity majority margins', in *Advanced Data Mining and Applications*, eds., Jie Tang, Irwin King, Ling Chen, and Jianyong Wang, volume 7120 of *Lecture Notes in Computer Science*, pp. 15–28. Springer, (2011).

[5] O. Cailloux, C. Lamboray, and Ph. Nemery, 'A taxonomy of clustering procedures', in *Proceedings of the 66th Meeting of the European Working Group on MCDA*, (2007).

[6] Y. De Smet and S. Eppe, 'Relational multicriteria clustering: The case of binary outranking matrices', in *Evolutionary Multi-Criterion Optimization. Fifth international conference, EMO 2009. Proceedings*, ed., M. et al. Ehrgott, volume 5467 of *Lecture Notes in Computer Science*, pp. 380–392. Springer Berlin, (2009).

[7] Y. De Smet and L. Guzman, 'Towards multicriteria clustering: an extension of the k-means algorithm', *European Journal of Operational Research*, **158**(2), 390–398, (2004).

[8] E. Fernandez, J. Navarro, and S. Bernal, 'Handling multicriteria preferences in cluster analysis', *European Journal of Operational Research*, **202**(3), 819 – 827, (2010).

[9] J. Figueira, V. Mousseau, and B. Roy, 'Electre methods', *Multiple criteria decision analysis: State of the art surveys*, 133–153, (2005).

[10] A. Jain, M. Murty, and P. Flynn, 'Data clustering: A review', *ACM Computing Survey*, **31**(3), 264–323, (1999).

[11] R.L. Keeney and H. Raiffa, *Decisions with multiple objectives: Preferences and value tradeoffs*, J. Wiley, New York, 1976.

[12] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, 'Optimization by simulated annealing', *Science*, **220**(4598), 671–680, (1983).

[13] P. Meyer and A-L. Olteanu, 'Preferentially ordered clustering', in *MDAI*, (2012).

[14] P. Nemery and Y. De Smet, 'Multicriteria ordered clustering', Technical Report TR/SMG/2005-003, Universite Libre de Bruxelles/SMG, (2005).

[15] A. Ngo The and V. Mousseau, 'Using assignment examples to infer category limits for the ELECTRE TRI method', *JMCDA*, **11**(1), 29–43, (2002).

[16] Clara Rocha, Luis C. Dias, and Isabel Dimas, 'Multicriteria classification with unknown categories: A clusteringsorting approach and an application to conflict management', *Journal of Multi-Criteria Decision Analysis*, (2012).

[17] B. Roy, 'Classement et choix en présence de points de vue multiples (la méthode electre)', *Revue française d'Informatique et de Recherche Opérationnelle (RIRO)*, **2**, 57–75, (1968).

[18] B. Roy and D. Bouyssou, *Aide Multicritère à la Décision : Méthodes et Cas*, Economica, Paris, 1993.

[19] Ph. Vincke, *Multicriteria Decision-Aid*, J. Wiley, New York, 1992.

[20] W. Yu, *Aide multicritère à la décision dans le cadre de la problématique du tri: méthodes et applications*, Ph.D. dissertation, LAMSADE, Université Paris Dauphine, Paris, 1992.

Learning the parameters of a multiple criteria sorting method from large sets of assignment examples

Olivier Sobrie[1,2,1] and **Vincent Mousseau**[2] and **Marc Pirlot**[3]

Abstract. ELECTRE TRI is a sorting method used in multiple criteria decision analysis. It assigns each alternative, described by a performance vector, to a category selected in a set of pre-defined ordered categories. Consecutive categories are separated by a profile. In a simplified version proposed and studied by Bouyssou and Marchant and called MR-Sort, a majority rule is used for assigning the alternatives to categories. Each alternative a is assigned to the lowest category for which a is at least as good as the lower profile delimiting this category for a majority of weighted criteria. In this paper, a new algorithm is proposed for learning the parameters of this model on the basis of assignment examples. In contrast with previous work ([8]), the present algorithm is designed to deal with large learning sets. Experimental results are presented, which assess the algorithm performances with respect to issues like model retrieval, computational efficiency and

1 Introduction

ELECTRE TRI is a sorting method used in decision analysis to assign each alternative to a category. The categories are pre-defined and ordered. A simplified version, called MR-Sort (Majority Rule Sorting method) has been studied by Bouyssou and Marchant (see [2, 3]). Alternatives are assigned to a category based on a majority rule. Each category is associated a lower profile defining its boundary with the category below. An alternative is assigned to one of the categories above a profile as soon as its performances are at least as good as those of the profile for a weighted majority of criteria.

Methods for eliciting the parameters of such a sorting method on the basis of assignment examples already exist but are limited to relatively small datasets. The question we are interested in is whether it is possible to use such rules in the context of preference learning, assuming that the learning datasets consist of a large number of assignment examples. For instance, the dataset can be composed of students' grades (satis bene, cum laude, magna cum laude, summa cum laude) corresponding to their results in the different disciplines. The goal is then to learn a MR-Sort model that assigns a grade to a student whenever a vector of his/her results in the different courses is entered. Learning such a model amounts to compute the profiles limiting the categories, the criteria weights and the majority threshold on the basis of a list of students, their marks and the grade they have been assigned to by the jury.

In [8], learning all the parameters of the MR-Sort has been formulated as a mixed integer linear program This formulation has a drawback: it is not suitable for large learning sets since it requires computing times that grow rapidly with the number of assignment examples.

This paper presents a metaheuristic we devised to infer all the parameters of an MR-Sort model. It reports the results of experiments testing the following aspects of the algorithm performance:

Model retrieval Given a set of alternatives assigned by a known MR-Sort model, what is the ability of the algorithm to determine the parameters of a model assigning these alternatives as much as possible to the same categories as the original model?

Algorithm efficiency What is the practical complexity of the algorithm? Is it able to deal with large learning sets? How much time does it take to learn the parameters of a model for a given number of categories, criteria and assignment examples?

Tolerance for error The learning set given as input to the algorithm might contain errors, e.g. an alternative that should belong to some category considering its evaluations could be erroneously assigned to a different category. The question is: How does the algorithm react to learning sets that are not entirely compatible with a MR-Sort model? Has the algorithm the ability to correct assignment errors?

In the next section of this paper, we briefly recall the rules of the MR-Sort procedure and which difficulties are involved in the elicitation of its parameters. We also discuss previous work done in view of eliciting the parameters of the MR-Sort rule. In section 3, we present the new metaheuristic. The experiments designed for testing it are described in section 4; our first experimental results are commented. We conclude with some perspectives for further work in view of improving the current version of the algorithm.

[1] email: olivier.sobrie@gmail.com

[2] École Centrale Paris, Grande Voie des Vignes, 92295 Châtenay Malabry, France, email: vincent.mousseau@ecp.fr

[3] Université de Mons, Faculté Polytechnique, 9, rue de Houdain, 7000 Mons, Belgium, email: marc.pirlot@umons.ac.be

2 Sorting procedure

2.1 MR-Sort model

The ELECTRE TRI procedure, originally developed in [14] (see also [13]), aims to assign every alternative a_i of a set $A = \{a_1, ..., a_n\}$ to one of the pre-defined and ordered categories going from C_1 to C_p, with C_1 the worst one and C_p the best one. Alternatives are evaluated on a set of n criteria; $a_{i,j}$ denotes the performance of a_i on criterion j. The criterion scales are assumed to be ordered in increasing order of the decision maker's preference. The assignment to a category is done by comparing each alternative performances to the performances of the $p-1$ profiles, delimiting the p categories, on each criterion. The profiles are denoted by b_h, $h = 1, \ldots, p-1$ and the performance of profile h on criterion j is $b_{h,j}$. The lower boundary of category C_h is profile b_{h-1}. For notational convenience, we sometimes use two trivial profiles b_0 and b_p. b_0 (resp. b_p) is the lower (resp. upper) profile of category C_1 (resp. C_p). For all j, the performance of b_0 on criterion j is the worst possible performance on this criterion, so that every alternative is at least as good as b_0 on all criteria. b_p plays a symmetric role in the sense that b_h is at least as good as every alternative on all criteria.

It is assumed that the profiles dominate each other, i.e.:

$$b_{h-1,j} \leq b_{h,j} \leq b_{h+1,j} h = 1, \ldots, p-1; j = 1, \ldots, p-1. \quad (1)$$

The original procedure presents some drawbacks. In particular, it involves numerous parameters which may play interrelated roles. Although several papers have been devoted to learning the parameters of such a model [10, 11, 9, 12, 6, 5], it is not advisable to use this method for learning preferences on the basis of large sets of assignment examples. In this article we consider a version of ELECTRE TRI called the *non-compensatory sorting model.* It is based on the work of Bouyssou and Marchant who have established an axiomatic characterization of this model in the case of two [2] or more categories [3].

To describe the assignment rule, we need to recall the definition of an *outranking* relation. An alternative a_i outranks a profile b_h if and only if there is a sufficient coalition of (weighted) criteria for which a_i is at least as good as b_h on each criterion of the coalition, and there is no criterion on which a_i is *so much worse* than b_h that compensating this disadvantage is impossible. The idea that some large disadvantages cannot be compensated usually is called a *veto*; it precludes asserting that a_i outranks b_h. The "at least as good" relation S_j on criterion j can be defined for instance by:

$$a_i S_j b_h \Leftrightarrow a_{i,j} \geq b_{h,j} \quad (2)$$

The sufficient majority of criteria j on which $a_i S_j b_h$ needed to say that the alternative a_i outranks the profile b_h is determined by the majority (or concordance) threshold λ.

The veto relation V_j on criterion j can be defined as follows:

$$a_i V_j b_h \Leftrightarrow b_{h,j} < a_{i,j} - v_j(b_h), \quad (3)$$

where $v_j(b_h)$ is called the *veto threshold* w.r.t. profile b_h.

If the sum of the weights w_j of the criteria j for which a_i is at least as good as b_h is larger than or equal to λ, and if there is no criterion on which there is a veto, then a_i outranks b_h.

The global outranking relation S is defined by:

$$a_i S b_h \Leftrightarrow \sum_{j \in S(a_i, b_h)} w_j \geq \lambda \text{ and } [\text{Not}[b_h V_j a_i], \forall j \in F] \quad (4)$$
$$\text{with } S(a_i, b_h) = \{j \in F : a_i S_j b_h\}.$$

Note that we do not assume that the decision maker has a preference relation on the whole set of alternatives that could be represented by a majority rule. It is well known, since Condorcet, that such a rule may lead to relations that lack the transitivity property and may have cycles. We only assume that the decision maker sorts the alternatives in ordered categories as if he or she would compare alternatives to the profiles limiting the categories using a rule like 5. Since the profiles are supposed to dominate each other, there can be no conflict related to intransitivity like aSb_h but not aSb_{h-1}.

With ELECTRE TRI, there are two ways to determine to which category an alternative should be assigned: they are called the pessimistic and the optimistic approach. We only describe the pessimistic approach (the only one that was characterized in [2, 3]) since it is the one used in the algorithm described below. The pessimistic procedure consists in comparing a_i to the profiles b_k for $k = p-1, p-2, \ldots, 1$ successively; if b_h is the first profile such that $a_i S b_h$, the alternative a_i is assigned to the category C_{h+1}. If the alternative a_i doesn't outrank any profile, then it is assigned to the worst category, C_1.

In this paper, we consider models without vetoes. Hence the conditions for an alternative a_i to be assigned to category C_h can be expressed as follows:

$$\sum_{j \in S(a_i, b_{h-1})} w_j \geq \lambda \qquad \text{and} \qquad \sum_{j \in S(a_i, b_h)} w_j < \lambda \quad (5)$$

As in [8], we call a model assigning alternatives to a category using such a rule, a *Majority Rule Sorting Model* (MR-Sort).

2.2 Elicitation of ELECTRE TRI parameters

Several published articles deal with learning the parameters of a traditional ELECTRE TRI model. In [10], it is proposed to infer the whole set of parameters of an ELECTRE TRI model from assignment examples by using a nonlinear programming formulation. In [9], the authors describe a way to learn the weights of an ELECTRE TRI model with a linear program. Article [12] deals with the inference of the profiles from assignment examples. Once again a linear program is used. In [7], a genetic algorithm is developed in order to learn the whole set of parameters of a traditional ELECTRE TRI model.

Recently, in [8], a mixed integer linear program has been proposed to infer all the parameters of an MR-Sort model. The linear program has been tested with 10 to 100 examples of assignments, 3 to 5 criteria and 2 to 3 categories. The experiments made show that a large number of assignment examples is needed to retrieve a model that represents the preferences of the decision maker with a reasonable accuracy. However, with the mixed integer linear program proposed in [8], the computing time quickly grows with the number of assignments. For instance, learning a 2 categories/3 criteria model (involving 7 parameters) takes less than one second on average, while it takes 7 seconds for 3 categories/4 criteria models (13 parameters) and 23 seconds for 3 categories/5 criteria (16 parameters). These computing times have been obtained when learning the models on the basis of 100 assignment examples (without assignment errors). For learning sets up to 100 examples, without assignment errors, the computing time grows roughly linearly with the number of examples. However, when the learning sets involves assignment errors, i.e. when some of the examples have not been assigned according to a presupposed MR-Sort model, computing times increase quite significantly with the percentage of errors. For a 2 categories/3 criteria models, the time needed to learn a model correctly reproducing the assignments of as many alternatives as possible from a set of 100 examples goes from an average of 4 seconds, for learning sets involving a 5% error rate, to 20 seconds, for a 15% error rate (see the detailed experimental results in [8]).

In [4], three mixed linear programs are used to find a set of weights or profiles which reflect as much as possible the preferences of multiple decision makers.

Using the linear programs developed in [8] and [4] is not an option in our case because we want to deal with large numbers of assignment examples and models having more than 5 criteria and 2 categories. The new approach proposed below aims at dealing with such models.

3 Inferring the parameters of a MR-Sort model

In this section we detail a new algorithm that aims to learn the whole set of parameters of an MR-Sort model. Initially, a set of random profiles dominating one another is generated. The proposed algorithm is an instance of alternating optimization [1]. The algorithm performs alternatively the following two main steps:

1. Using the current profiles, find a set of weights and a majority threshold maximizing the number of assignment examples compatible with the model;
2. Adjust the profiles in order to maximize the number of assignment examples compatible with the model.

The goal of the algorithm is to obtain the parameters of a model reflecting as much as possible the preferences of the decision maker, i.e. a model that restores as much as possible the assignments of the examples given as input. To measure the performance of the algorithm, we compute the classification accuracy CA of the final model, which is defined as:

$$CA = \frac{\text{Number of examples correctly restored}}{\text{Total number of examples}} \tag{6}$$

In this section, we first describe the linear program used to learn the weights. Then we describe the metaheuristicused to improve the position of the profiles. Finally, the coupling of the linear program and the metaheuristicis explained.

3.1 Inferring the weights and the majority threshold

Finding the weights and the majority threshold of an MR-Sort model with fixed profiles doesn't require mixed integer programming. The problem can be easily formulated as a simple linear program.

We denote by A_h the set of alternatives assigned by the DM to the category C_h. As the profiles dominate each other, the constraints for an alternative a_i to be assigned to the category C_h can be expressed as follows:

$$\sum_{\forall j | a_i S_j b_{h-1}} w_j - x_i + x'_i = \lambda \qquad \forall a_i \in A_h, \quad h = \{2, ..., p-1\} \tag{7}$$

$$\sum_{\forall j | a_i S_j b_h} w_j + y_i - y'_i = \lambda - \delta \qquad \forall a_i \in A_h, \quad h = \{1, ..., p-2\} \tag{8}$$

with:

$$\sum_{j=1}^{n} w_j = 1 \tag{9}$$

$$\lambda \in [0.5; 1] \tag{10}$$

$$w_j \in [0; 1] \qquad \forall j \in F \tag{11}$$

$$x_i \in \Re_0^+ \qquad \forall a_i \tag{12}$$

$$y_i \in \Re_0^+ \qquad \forall a_i \tag{13}$$

$$x'_i \in \Re_0^+ \qquad \forall a_i \tag{14}$$

$$y'_i \in \Re_0^+ \qquad \forall a_i \tag{15}$$

The value $x_i - x'_i$ (resp. $y_i - y'_i$) represents the difference between the sum of the weights of the criteria belonging to coalition in favor of $a_i \in A_h$ w.r.t. b_{h-1} (resp. b_h) and the majority threshold. If both $x_i - x'_i$ and $y_i - y'_i$ are positive, then the alternative a_i is assigned to the right category. In order to try to get a maximum number of compatible alternatives, the objective function of the linear program minimizes the sum of x'_i and y'_i:

$$\min \sum_{a_i \in A} (x'_i + y'_i) \tag{16}$$

However this objective function does not guarantee to return a set of weights and a majority threshold which maximize the number of alternatives assigned to the category indicated by the DM. This is due to the fact that the objective function allows for compensatory effects between constraints.

3.2 Inferring the profiles

Trying to learn the profiles using an exact method is not easy because conditions (5) cannot be formulated as linear constraints. Exact methods have been proposed and studied in [8]. They require mixed integer programming solvers. As we want to deal with models having more than 5 criteria and 2 categories, the use of a linear program with binary variables is not an option due to quickly exploding computing times. Therefore we opted for developing a metaheuristic, which is described below.

3.2.1 Idea of the metaheuristic

Consider a model with 5 criteria, 2 categories, C_1 and C_2 (with $C_2 \succ C_1$). We assume that the criteria weights are known. Let a_1 and a_2 be 2 misclassified alternatives, (see figure 1). The profile delimiting the two categories is denoted by b_1; b_0 and b_2 correspond respectively to the worst and the best possible (fictive) alternative on the five criteria. Imagine that a_1 is wrongly assigned by the procedure to the category C_2 instead of C_1. This means that the profile has too low levels on one or several criteria. In contrast, an alternative a_2 wrongly assigned to category C_1 instead of C_2 means that the profile level is too high on one or several criteria (we recall that the weights are considered as known). On figure 1, an arrow shows the direction in which moving the profile in order to potentially assign the two alternatives to the right category.

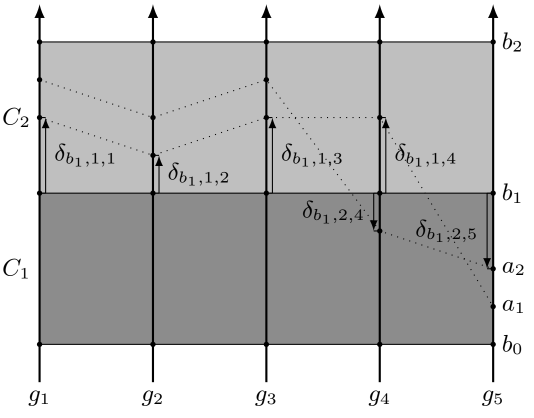


Figure 1. Alternative wrongly assigned because of the profile too low or too high

We denote by A_1^2 (resp. A_2^1) the set of alternatives wrongly classified in C_2 (C_1, respectively) by the inferred model while the DM assigns them to category C_1 (resp. C_2). The sets of alternatives correctly classified in C_1 and C_2 are denoted respectively by A_1^1 and A_2^2. With a two categories model, each alternative belongs to one of the four sets, A_1^1, A_2^1, A_1^2 or A_2^2. An alternative belonging to A_2^1 (resp. A_1^2) indicates that the profile level is too high (resp. too low) on one or several criteria (assuming that we do not change the weights).

Regarding the relative position of the evaluation $a_{i,j}$ of alternative a_i on criterion j and the profile evaluation $b_{1,j}$ and considering the assignment of the alternative, we can distinguish 8 cases (see figure 2); $\delta_{1,i,j}$ represents the distance between the profile level and the alternative evaluation on criterion j.

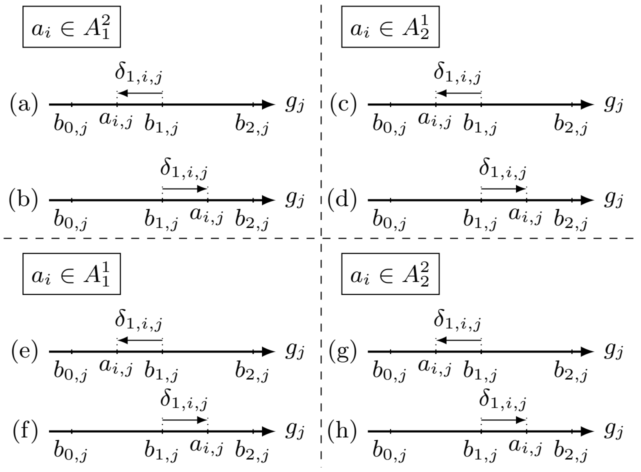

$a_i \in A_1^2$
$a_i \in A_2^1$
(a)
$\delta_{1,i,j}$
$b_{0,j}$ $a_{i,j}$ $b_{1,j}$ $b_{2,j}$
g_j
(c)
$\delta_{1,i,j}$
$b_{0,j}$ $a_{i,j}$ $b_{1,j}$ $b_{2,j}$
g_j
(b)
$\delta_{1,i,j}$
$b_{0,j}$ $b_{1,j}$ $a_{i,j}$ $b_{2,j}$
g_j
(d)
$\delta_{1,i,j}$
$b_{0,j}$ $b_{1,j}$ $a_{i,j}$ $b_{2,j}$
g_j
$a_i \in A_1^1$
$a_i \in A_2^2$
(e)
$\delta_{1,i,j}$
$b_{0,j}$ $a_{i,j}$ $b_{1,j}$ $b_{2,j}$
g_j
(g)
$\delta_{1,i,j}$
$b_{0,j}$ $a_{i,j}$ $b_{1,j}$ $b_{2,j}$
g_j
(f)
$\delta_{1,i,j}$
$b_{0,j}$ $b_{1,j}$ $a_{i,j}$ $b_{2,j}$
g_j
(h)
$\delta_{1,i,j}$
$b_{0,j}$ $b_{1,j}$ $a_{i,j}$ $b_{2,j}$
g_j

Figure 2. Given the evaluation of an alternative and of the profile on a criterion j, 8 possible cases regarding the alternative assignment

In the 8 cases represented in figure 2, we see that the difference between the value of the profile $b_{1,j}$ and the performance of the alternative $a_{i,j}$ can have a positive (cases a, d, e, h) or a negative (cases b, c, f, g) influence on the classification. We denote by $W_{1,j}$ the set of alternatives wrongly assigned by the model and for which the criterion j is not in favor of the correct assignment due to the current profile level. The set $R_{1,j}$ contains the alternatives for which evaluation of b_1 favors the assignment to the right class.

$$\begin{aligned} W_{1,j} &= \left\{a_i \in A_1^2 : a_{i,j} \geq b_{1,j}\right\} \\ &\quad \cup \left\{a_i \in A_2^1 : a_{i,j} < b_{1,j}\right\} \qquad (17) \\ R_{1,j} &= \left\{a_i \in A_1^2 : a_{i,j} < b_{1,j}\right\} \\ &\quad \cup \left\{a_i \in A_1^1 : a_{i,j} < b_{1,j}\right\} \\ &\quad \cup \left\{a_i \in A_2^1 : a_{i,j} \geq b_{1,j}\right\} \\ &\quad \cup \left\{a_i \in A_2^2 : a_{i,j} \geq b_{1,j}\right\} \qquad (18) \end{aligned}$$

The alternatives contained in the sets $W_{1,j}$ and $R_{1,j}$ give an indication about how the profile should be moved on criterion j to potentially increase the classification accuracy of the model. In order to assess the advantage of the different possible moves of the profile level, the space between the profiles levels $b_{1,j}$ and $b_{0,j}$ on criterion j is split into k sub-intervals by means of k subdivision points denoted by $b_{1,j}^{-l}$ for $l = 1, \ldots, k$. The same is done between $b_{1,j}$ and $b_{2,j}$ by means of k subdivision points denoted by $b_{1,j}^{+l}$ for $l = 1, \ldots, k$. We consider these $2k$ subdivision points scattered on both sides of $b_{1,j}$ as the candidate moves for the profile level $b_{1,j}$. Then, histograms similar to those shown in figure 3 are constructed for each criterion j.

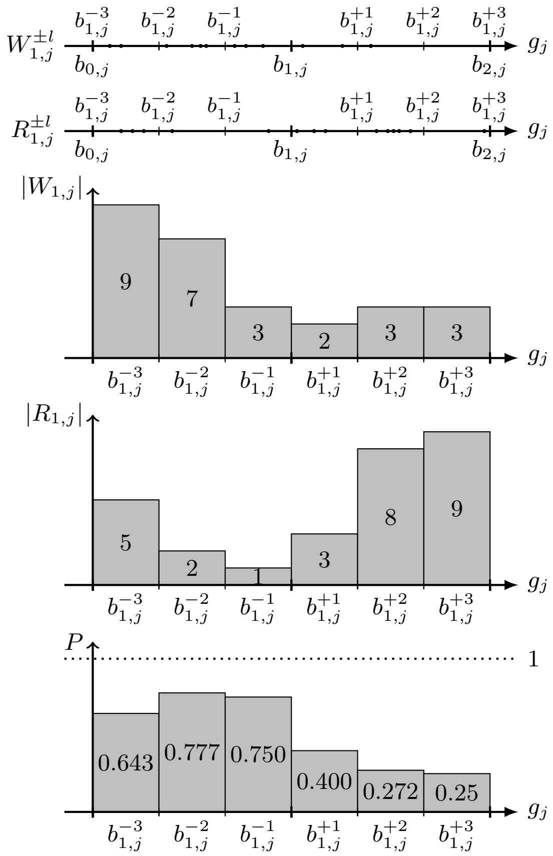


Figure 3. Histogram of the evaluations of misclassified alternatives on criterion j

The bars lengths in the first (resp. second) histogram represent the number of alternatives in the set $W_{1,j}^{\pm l}$ (resp. $R_{1,j}^{\pm l}$) where $W_{1,j}^{\pm l}$ (resp. $R_{1,j}^{\pm l}$) denotes the set of alternatives belonging to $W_{1,j}$ (resp. $R_{1,j}$) the evaluation of which, on criterion j, is located between the current value of the profile, $b_{1,j}$, and the potential new value, $b_{1,j}^{\pm l}$. In the last histogram, the bars lengths represent what is formally a probability P defined by:

$$P(b_{1,j}^{\pm l}) = \frac{|W_{1,j}^{\pm l}|}{|W_{1,j}^{\pm l}| + |R_{1,j}^{\pm l}|}| \tag{19}$$

If we move the profile level $b_{1,j}$ to $b_{1,j}^{\pm l}$, the number $|W_{1,j}^{\pm l}|$ will decrease by $|W_{1,j}^{\pm l}| - |R_{1,j}^{\pm l}|$ and the number $|R_{1,j}^{\pm l}|$ will increase by the same quantity. If the quantity $|W_{1,j}^{\pm l}| - |R_{1,j}^{\pm l}|$ is positive, the number of correctly assigned alternatives with their evaluation on the right side of the profile will tend to increase while the profile level is moved to $b_{1,j}^{\pm l}$. Of course, the number of correctly assigned alternatives will not mechanically increase by $|W_{1,j}^{\pm l}| - |R_{1,j}^{\pm l}|$ since the corresponding change in the profile level only concerns criterion j. We use the probabilities $P(b_{1,j}^{\pm l})$ as indicators of the potential gain in correct classification that can be expected from a move of the profile level on some criterion. The probabilities associated with profile b_1 on criterion j are computed and the value $L \in \{-k, \ldots, -1, 1, \ldots, k\}$ for which the probability of $b_{1,j}^{L}$ is maximal is recorded. Then a random number r is drawn from the uniform distribution on $[0, 1]$. If the value of r is smaller than $P(b_{1,j}^{\pm L})$, then the profile is moved to $b_{1,j}^{\pm L}$, otherwise the profile is not moved at all. The same operation is performed for each criterion.

One loop of the metaheuristicin the case of a model with 2 categories can be summarized by the following algorithm:

```
for all $j \in \{1, ..., n\}$ do
  Compute $P(g_j(b_1^{\pm l}))$, $\forall l$
  Find $L$ such that $P(g_j(b_1^L)) = \max_l(P(g_j(b_1^{\pm l})))$
  Draw a random number $r$ from the uniform distribution [0, 1]
  if $r < (P(b_{1,j}^L))$ then
    $b_{1,j} = b_{1,j}^L$
  end if
end for
```

When there are more than two categories, a similar algorithm is applied to each profile with a slightly adapted definition of W and R:

$$\begin{aligned} W_{h,j} &= \Big\{a_i \in A_h^{h+1} : b_{h+1,j} > a_{i,j} \geq b_{h,j}\Big\} \\ &\cup \Big\{a_i \in A_{h+1}^{h} : b_{h-1,j} < a_{i,j} < b_{h,j}\Big\} \end{aligned} \tag{20}$$

$$\begin{aligned} R_{h,j} &= \Big\{a_i \in A_h^{h+1} : b_{h-1,j} \leq a_{i,j} < b_{h,j}\Big\} \\ &\cup \Big\{a_i \in A_h^{h} : b_{h-1,j} \leq a_{i,j} < b_{h,j}\Big\} \\ &\cup \Big\{a_i \in A_{h+1}^{h} : b_{h+1,j} > a_{i,j} \geq b_{h,j}\Big\} \\ &\cup \Big\{a_i \in A_{h+1}^{h+1} : b_{h+1,j} > a_{i,j} \geq b_{h,j}\Big\} \end{aligned} \tag{21}$$

for $h = 1, ..., p-1$. In these definitions, A_h^l denotes the subset of misclassified alternatives that are assigned to category C_l by the model while the DM assigns them to category C_k. Note that the definitions of $W_{h,j}$ and $R_{h,j}$ only take into account the alternatives for which the class assigned by the DM and the model either coincide or are nearest neighbor. Definitions which take into account all misclassified alternatives have been experimented and have led to inferior results in terms of convergence of the algorithm.

3.2.2 Parameters setting and tactical details

In the metaheuristicoutlined above, several parameters and implementation details influence the convergence. The following options have been chosen.

Objective function and stopping criterion In the proposed algorithm, the objective function aims at maximizing the classification accuracy of the model. The stopping criterion is met once the classification accuracy is equal to 1 or when the algorithm has run for N_{it} loops.

Number and position of the subdivision points $b_{i,j}^{\pm l}$ The interval in which the value of the profile b_h can vary is subdivided $2k$ subintervals. The number k and the way of subdividing the interval (equal vs. unequal subintervals) must be specified.

Probability function In the present version, the probability (19) only takes into account the number of alternatives rightly or wrongly assigned to one of the two categories neighboring the profile.

Treatment order of the profiles When there are more than two categories, we have to specify the order in which the algorithm handles the profiles. In this paper, they are treated in ascending order of their labels, i.e. b_1, b_2, ..., b_{p-1}.

3.3 Inferring all the parameters

To infer all the parameters of the MR-Sort procedure, the linear program and the metaheuristic, described in the previous paragraphs, are combined.

First, a set of N_{mod} MR-Sort models is generated. Each model is initialized with a set of random profiles. Then, for each model, the following two operations are repeated at most N_o times:

1. Given the current profiles, the weights and a majority threshold are learned by using the linear program.
2. Given the current values of the weights and the threshold, the profiles are improved by running the metaheuristicN_{it} times. The classification accuracy CA, is computed after each loop. After the N_{it} loops, the profiles giving the best CA are kept.

After the 2 steps learning procedure has been applied to the N_m models, the algorithm keeps only the $N_m/2$ models giving the best CA and $N_m/2$ new models are randomly generated. The algorithm is stopped once a model has a CA equal to 1 or when the algorithm has run N_o times.

4 Experimentations

In this section, we address the validation issues presented in the introduction, i.e. model retrieval, algorithm efficiency and tolerance for errors. We successively test the linear program used to infer the weights and the majority threshold, the metaheuristic used to infer the profiles and the metaheuristic allowing to infer the whole set of parameters.

To test the algorithm partially and globally, we use a common testing procedure:

1. A random MR-Sort model M is generated. It is determined by a set of weights, normalized to 1, a set of profiles ordered by the dominance relation with evaluations on the n criteria between 0 and 1 and a majority threshold whose value is picked in the interval $[0.5, 1]$. All values are drawn from uniform distributions. The $p-1$ profiles are generated by drawing $p-1$ numbers at random on each criterion. These numbers are reordered in increasing order. The number of rank h on criterion j is the jth component $b_{h,j}$ of profile b_h. Using model M as described by (5), each alternative can be assigned to a category. The resulting assignment rule is referred to as s_M.
2. A set of m alternatives with random performances on the n criteria is generated. The performance values are drawn uniformly and independently from the $[0, 1]$ interval. The set of generated alternatives is denoted by A. The alternatives in A are assigned using the rule s_M. The resulting assignments and the performances of the alternatives in the set A are given as input to the algorithm. They constitute the learning set.
3. In case we only infer part of the parameters of the rule, the other parameters are given as input to the algorithm, e.g. for the inference of the profiles, the weights and the majority threshold are given as input. Then the algorithm runs and tries to maximize the number of assignments compatible with the output resulting from step 2. The resulting model is denoted by M'. The alternatives in the set A are assigned to a category by the model M'. Formally, the assignment rule is denoted by $s_{M'}$. We compute the classification accuracy $CA(s_M, s_{M'})$ according to equation 6, i.e. $CA(s_M, s_{M'}) = \frac{|\{a \in A : s_M(a) = s_{M'}(a)\}|}{|A|}$.

The last step allows to study the efficiency of the algorithm either by examining the computing time needed to learn the parameters or by observing the algorithm convergence behavior. In order to answer to the two other questions posed in the introduction, additional steps are required:

4. After learning the parameters, a set of 10000 alternatives with random performances (drawn independently from the uniform distribution in the $[0, 1]$ interval) is generated. It is denoted by B. This set is used in the generalization phase.
5. The alternatives contained in the set B are assigned using rules s_M and s'_M and the classification accuracy of model M' is computed.

These two steps allow us to address the model retrieval issue. To see how the algorithm behaves when the learning set contains errors, an additional step is needed:

2' A proportion of error is added in the assignment resulting from rule s_M. We denote by $\tilde{s}_M$ the rule producing the assignments with errors.

After learning the parameters of the MR-Sort model, two values of classification accuracy can be computed to analyze the algorithm behavior in the presence of errors. On the one hand, the value $CA(s_M, s_{M'})$ gives an indication on the ability of the algorithm to correct the errors in assignment examples. On the other hand, the value $CA(\tilde{s}_M, s_{M'})$ gives an indication on the ability of the algorithm at finding a model fitting the learning set given as input.

The experimentations presented are made on an Intel Core 2 P8700 PC running Gentoo Linux, CPLEX version 12 and

Python 2.7.2. All experiments are repeated on 10 random instances and the values displayed in the graphics below are averages over these 10 instances.

4.1 Inference of the weights and majority threshold

4.1.1 Computing time

To see how much time is needed to learn the weights and the majority threshold, the linear program, described in subsection 3.1 is tested for models with 3 categories and 5, 7, 10 or 20 criteria with 1000 to 10000 assignment examples. The profiles that are used are the correct ones, i.e. those used in the rule s_M that assigns the alternatives in the learning set.

Solving large continuous variables linear programs using a solver like CPLEX can be done very efficiently. However, a pre-treatment of the linear constraints is required in order to reduce the computing time needed to encode the constraints into the solver. The pre-processing consists in consists in filtering the constraints 7 in view of eliminating the redundant ones

The experimental results are displayed in Figure 4. It shows that less than 1 second is needed to learn the parameters of a model having 3 categories and 10 criteria, even when the learning set is as large as 10000 alternatives. However we see that the computing time increases with the number of criteria. This is due to the fact that the number of non-redundant constraints quickly grows when the number of criteria is increased.

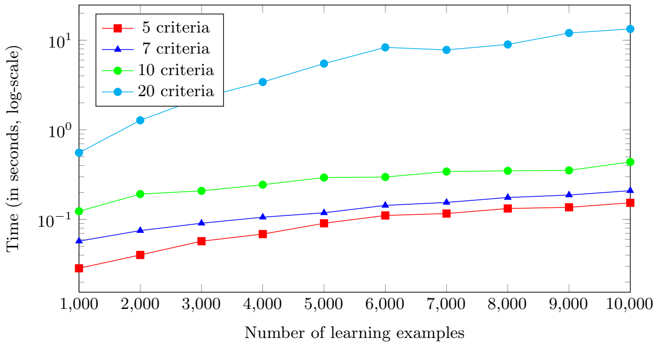


Figure 4. Computing time for learning the weights and the majority threshold of a model with 3 categories and 5, 7, 10 or 20 criteria

4.1.2 Model retrieval

What is the number of alternatives needed to obtain a good set of weights and majority threshold for a model with a given number of categories and criteria (assuming that we start with the right profiles)?

The algorithm is tested on 3 categories and 10 criteria models with learning sets involving 100 to 1000 assignment examples. The inferred model ($s_{M'}$) is used "in generalization" to assign 10000 randomly generated alternatives. These assignments are compared with those made by the original rule (s_M), yielding an assessment of the classification accuracy. The evolution of the classification accuracy is shown on figure 5.

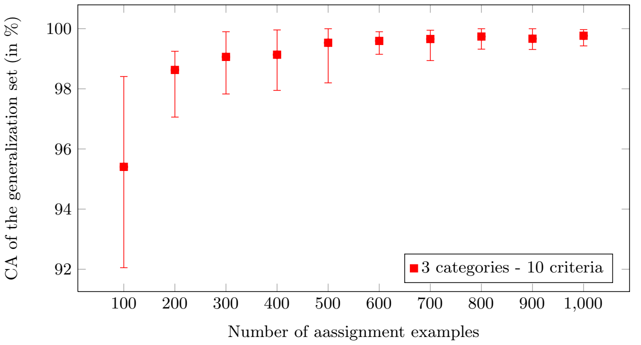


Figure 5. Evolution of the classification accuracy of models having 3 categories and 10 criteria when the learning set contains 100 to 1000 alternatives

As we can see from the plot, the linear program returns weights and a threshold that allow to assign the alternatives in a similar way as the original model even for relatively small learning sets. The classification accuracy is above 95 % for 200 assignment examples; it quickly reaches a classification accuracy close to 100 % when the number of alternatives increases.

4.1.3 Tolerance for error

Starting with learning sets in which the alternatives have been assigned according to rule s_M, we introduce random assignment errors. More precisely, a certain proportion of the alternatives are reassigned to another category chosen uniformly at random among all the other categories. We investigate how the algorithm reacts.

The algorithm for learning the weights and a threshold is tested on 3 categories and 10 criteria models when a proportion of 5 to 40 % of assignment errors are introduced in learning sets composed of 1000 assignment examples. Once the parameters have been learned, we compare the original model and the learned one on the manner they assign the alternatives in the learning set.

Figure 6 displays the average, minimal and maximal values of the classification accuracy obtained in the generalization set (10000 alternatives) on the basis of learning sets containing 5 to 40 % of erroneous assignments. Since the number of assignment errors made by the learned model usually is smaller than the number of assignment errors introduced in the learning set, we conclude that the algorithm selects weights and a threshold in such a way that some of the errors introduced in the learning set are corrected, thus obtaining a classification accuracy $CA(s_M, s_{M'})$ that is generally better than 100 % minus the assignment error rate in the learning set. When using the learned model to assign alternatives in a generalization set, the error rate usually is smaller than that in the learning set.

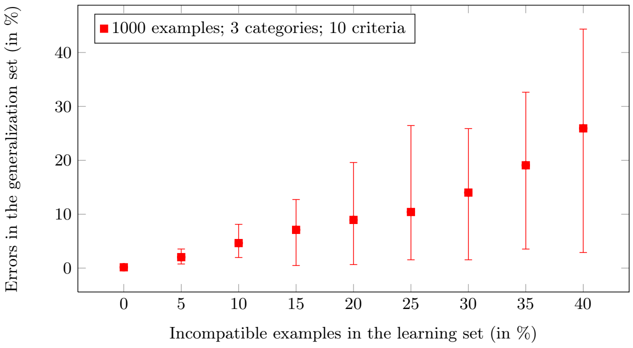


Figure 6. Evolution of the number of assignment errors made by the learned model for the alternatives in the generalization set (10000 alternatives). The original model has 3 categories and 10 criteria and the learning set contains 5 to 40 % of erroneous assignments

A further issue is the following. Are the alternatives in the learning set wrongly assigned by the learned model mostly alternatives that have been erroneously reassigned to introduce errors in the learning set? Or, on the opposite, does the learned model create many new assignment errors? In the set of alternatives wrongly assigned with the learned weights and majority threshold, what is the percentage of alternatives that were degraded in this set? By looking at the set of alternatives incorrectly assigned by the function $s_{M'}$, we see that these alternatives are mainly ones that were not errors. For instance, in a case in which the learning set is composed of 1000 alternatives, erroneously assigned for 10% of them, among the 5% of errors obtained by assigning the alternatives of the learning set by means of M', only 0.5% correspond to errors introduced in the learning set. We conclude that the algorithm is able to correct introduced assignment errors, but will in general create other errors.

4.2 Inference of the profiles

4.2.1 *Strategy for moving the profiles*

As emphasized in section 3.2.2, the convergence of the algorithm is influenced by several parameters. Among these, we now focus on the size of the intervals and the number of intervals determining the possible moves for the profiles. We present the evolution of the classification accuracy in connection with 3 different strategies for defining the potential profiles moves.

1. Equally spaced subdivisions between the profiles.

$$b^{+l}_{h,j} = b_{h,j} + \frac{l}{k} \cdot (b_{h+1,j} - b_{h,j}) \tag{22}$$

$$b^{-l}_{h,j} = b_{h,j} - \frac{l}{k} \cdot (b_{h,j} - b_{h-1,j}) \tag{23}$$

   with k the number of sub-intervals and $l \in \{1, ...k\}$.

2. Spacing between two subdivisions increasing as a function of the distance to the profile.

$$b^{+l}_{h,j} = b_{h,j} + \frac{e^l}{\sum_{i=1}^{k} e^i} \cdot (b_{h+1,j} - b_{h,j}) \tag{24}$$

$$b^{-l}_{h,j} = b_{h,j} - \frac{e^l}{\sum_{i=1}^{k} e^i} \cdot (b_{h,j} - b_{h-1,j}) \tag{25}$$

3. Spacing between subdivisions increasing as a function of the distance to the profile; Number of intervals increasing as a function of the classification accuracy of the model.

We see on figure 7 that the third strategy guarantees a faster convergence. It is the one that is used in the rest of the experimentations.

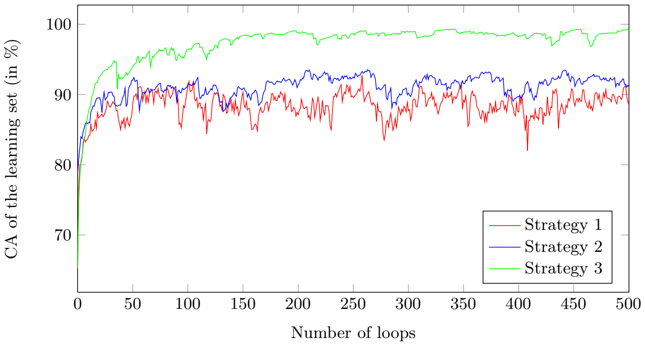


Figure 7. Evolution of the classification accuracy for 3 categories and 10 criteria models, depending on the strategy adopted for moving the profiles

4.2.2 *Model retrieval*

The experiments are made on models having 3 categories and 10 criteria. Using the right weights and threshold (those of model M), the profiles are learned on the basis of learning sets involving from 100 up to 1000 assignment examples. The resulting model M' is then used to assign 10000 randomly generated alternatives. Their assignment by M' is then compared with their assignment using M. The average, minimal and maximal values of the classification accuracy for the 10 instances are plotted on figure 8.

With 1000 alternatives in the learning set, the classification accuracy of the alternatives contained in the generalization set is on average close to 100 %. Unlike the linear program used to find the weights and the majority threshold, the metaheuristic requires more examples to return an appropriate set of profiles. This can be explained on the one hand by the number of parameters to be determined is larger (when there are more than two categories) and on the other hand by the fact that the metaheuristic can remain stuck in a local minimum.

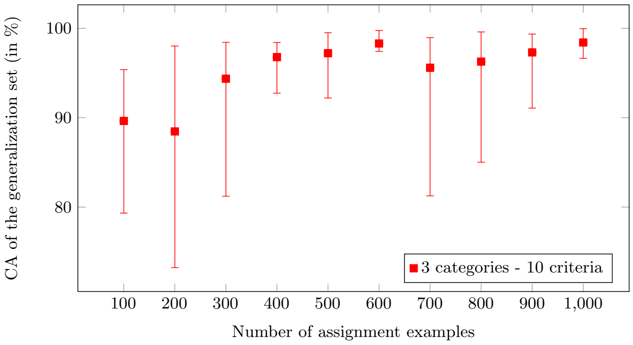


Figure 8. Evolution of the classification accuracy in generalization (10000 alternatives) for models having 3 categories and 10 criteria; size of learning set: 100 up to 1000 alternatives

4.2.3 Tolerance for error

Experiments are made on models having 3 categories and 10 criteria; the learning set involves 1000 alternatives. A proportion of random erroneous reassignments is applied to the learning set. Model M' is learned on the basis of this corrupted learning set and then, the assignments of the alternatives in the learning set by model M' are compared to those produced by the corrupted rule $\tilde{s}_M$. Figure 9 indicates that the classification accuracy of the learning set with errors, $CA(\tilde{s}_M, s_{M'})$, *tends to be higher than the percentage of errors introduced in the learning set. signification ????*

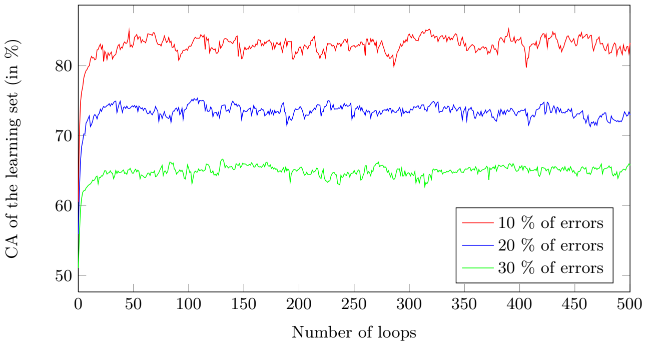


Figure 9. Evolution of the classification accuracy w.r.t. the erroneous learning set ($CA(\tilde{s}_M, s_{M'})$) used to learn the profiles of models having 3 categories and 10 criteria with 1000 learning alternatives containing 10 to 30 % of incompatible assignments

In the case of learning sets with 10% of introduced errors, the classification accuracy $CA(\tilde{s}_M, s_{M'})$ is more or less equal to 85% after 50 loops of the algorithm. Looking at the 15% of alternatives erroneously assigned, we observe that 9.5% are alternatives that have been reassigned (i.e. belong to the assignment errors introduced in the learning set). This indicates that the algorithm has the ability to identify wrong assignments and adjust the parameters on the basis of the learning examples which are not corrupted. However, the algorithm also introduces some additional errors while assigning the alternatives in the learning set.

The observations just made let us expect good results in generalization, as the learned model M' seems to be close to the original–uncorrupted–model M. To challenge this feeling, we compare the assignments obtained by the learned model M' and the original model M on a set of 10000 randomly generated alternatives.

Figure 10 shows that the metaheuristic has a good capacity to retrieve assignment examples even in the presence of errors in the learning set. For instance, with 10 % of errors in the learning set, the model learned by means of the algorithm correctly assigns 97.5 % of the alternatives in the generalization set.

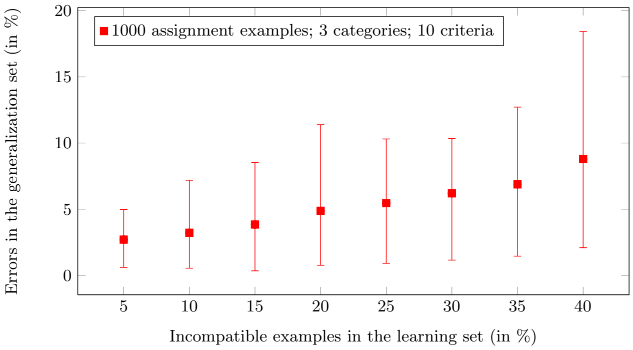


Figure 10. Evolution of the classification errors in the generalization set (10000 alternatives) after learning the profiles of models having 3 categories and 10 criteria on the basis of a set of 1000 assignment examples containing 5 to 40 % assignment errors

4.3 Inference of all parameters

For the inference of the whole set of parameters, the same experiments as for the partial inferences have been performed.

4.3.1 Convergence of the algorithm

Our first concern is to study the convergence behavior of the combined algorithm. The program described for the inference of all parameters of a MR-Sort model is tested for models having 3 categories and 10 criteria. The algorithm is run 100 times ($N_o = 100$) on a population of 10 models ($N_{mod} = 10$). For each loop, the linear program is run once and the metaheuristic 20 times (N_{it}).

Figure 11 displays the average classification accuracy of the alternatives in the learning set after each loop. We observe that the strongest improvement in the classification accuracy is obtained during the first iteration. It is then not needed to run the algorithm for too long because the gain in classification accuracy will not be substantial. In the example presented, 10 iterations of the combined algorithm appears to be sufficient.

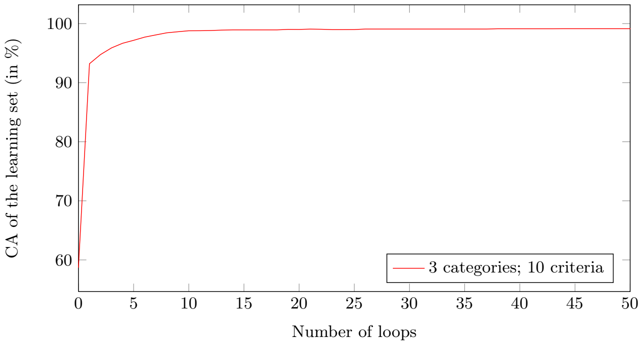


Figure 11. Evolution of the classification accuracy of the alternatives in the learning set used to learn the profiles of a model having 3 categories and 10 criteria with 1000 assignment examples ($N_{mod} = 10; N_o = 100; N_{it} = 20$)

4.3.2 Model retrieval

How many examples should we consider in the learning set in order to be able to infer a model that gives a fair representation of the decision maker's preferences? The experimentation is performed for 3 categories and 10 criteria models. The learning sets involve from 100 up to 1000 assignment examples. We study the classification accuracy of the learned model M' by comparing the assignments of 10000 randomly generated alternatives. Figure 12 shows the classification accuracy $CA(s_M, s_{M'})$ in generalization.

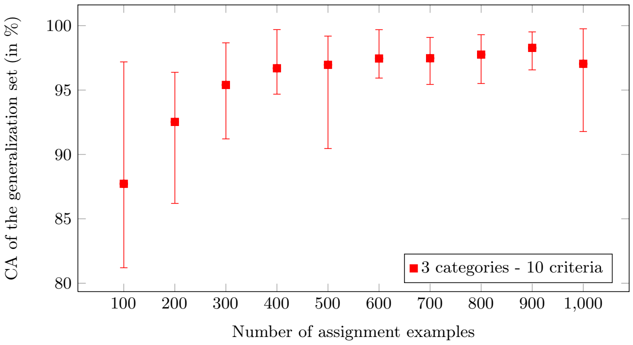


Figure 12. Evolution of the classification accuracy on the generalization set. A 3 categories and 10 criteria model has been learned on the basis of learning sets containing from 100 up to 1000 assignment examples ($N_{mod} = 10; N_o = 100; N_{it} = 20$)

The plot shows us that the learned model is tuned precisely enough to produce a classification accuracy superior to 90 % on average when at least 300 assignment examples are used. As expected, the larger the number of assignment examples used as input, the more precise the model.

4.3.3 Tolerance for error

We also want to know the capacity of the algorithm to return appropriate values for the parameters when the learning set contains erroneous assignments. The experimental setting is the same as before.

Using the learned model M' to assign the examples in the learning set yields the results displayed on figure 13.

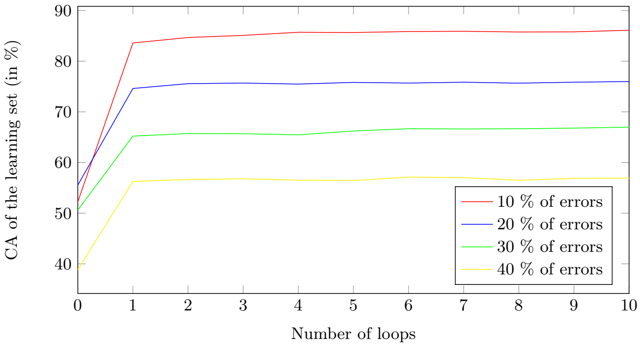


Figure 13. Evolution of the classification accuracy ($CA(\tilde{s}_M, s_{M'})$) for the alternatives in the learning set. A 3 categories and 10 criteria model is inferred on the basis of 1000 assignment examples containing 10 to 30 % of errors ($N_{mod} = 10; N_o = 100; N_{it} = 20$)

We see that the classification accuracy reflects the percentage of errors in the learning set. For 10 % of errors in the learning set given in input, the classification accuracy of the learning set after learning the model stays between 85 % and 90 %.

To assess the ability of the algorithm to identify incorrectly assigned alternatives, we study its behavior in generalization by randomly generating 10000 alternatives and comparing their assignment by the original model M and the learned model M'. The results are shown on figure 14.

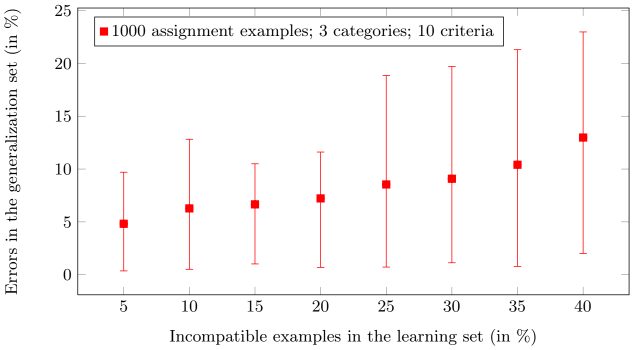


Figure 14. Evolution of the classification errors on the generalization set (10000 alternatives) after learning the whole set of parameters on the basis 1000 assignment examples ($N_{mod} = 10; N_o = 100; N_{it} = 20$)

The percentage of errors in the learning set is on average attenuated by the metaheuristic. For instance with 20 % of errors in the learning set, the average error is around 8 % in the generalization set with the learned model. However, the metaheuristic sometimes returns models producing a percentage of assignment errors superior to the error rate imposed on the learning set. For instance, with 5 % of errors in the learning set, the metaheuristic returned once 10 % of errors in the generalization set. This demonstrates that there is still room for improving the algorithm, which may currently fail to converge to a learned model sufficiently close to the original one.

5 Conclusion and further research issues

In this article we presented a new metaheuristic to learn the whole set of parameters of an MR-Sort procedure. Several experiments have been performed for testing the behavior of the metaheuristic when large learning sets are used as input.

In the experimentations, we observed that we can obtain good results for reasonably complex models i.e. typically those involving 3 categories and 10 criteria. The metaheuristic can retrieve the parameters of such models from 500 assignment examples with a classification accuracy close to 95 %. When the assignment of the examples in the learning set is not fully compatible with a MR-Sort model, the metaheuristic is still able to return a reasonable approximation of the “true” model by an adequate MR-Sort model. In generalization, we saw that the percentage of assignment errors made by the learned model is smaller than the percentage of assignment errors introduced the learning set.

However, the experiments have also shown that additional work is needed to improve the algorithm behavior in some cases. When there are no assignment errors in the learning set *there are assignment errors ?????* , it happens that the metaheuristic used for learning the profiles does not converge towards a classification accuracy of 100 % even after more than 500 loops. Several tactical options for implementing the metaheuristic have been presented in section 3.2.2 but only two of them have been studied. Other parameters like the probability function used for choosing the profiles moves deserve to be studied in the perspective of improving the algorithm performance.

This paper does not cover the case of a MR-Sort model with vetoes as described in section 2.1. Learning the parameters of such model is another challenge.

REFERENCES

[1] Bezdek, J.C., Hathaway, R.J.: Some notes on alternating optimization. In: Pal, N.R., Sugeno, M. (eds.) Advances in Soft Computing, AFSS 2002, Lecture Notes in Computer Science, vol. 2275, pp. 288–300. Springer Berlin Heidelberg (2002)

[2] Bouyssou, D., Marchant, T.: An axiomatic approach to noncompensatory sorting methods in MCDM, I: The case of two categories. European Journal of Operational Research 178(1), 217–245 (2007)

[3] Bouyssou, D., Marchant, T.: An axiomatic approach to noncompensatory sorting methods in MCDM, II: More than two categories. European Journal of Operational Research 178(1), 246–276 (2007)

[4] Cailloux, O., Meyer, P., Mousseau, V.: Eliciting ELECTRE TRI category limits for a group of decision makers. European Journal of Operational Research (2012), in press.

[5] Dias, L., Mousseau, V.: Inferring Electre's veto-related parameters from outranking examples. European Journal of Operational Research 170(1), 172–191 (2006)

[6] Dias, L., Mousseau, V., Figueira, J., Clímaco, J.: An aggregation/disaggregation approach to obtain robust conclusions with ELECTRE TRI. European Journal of Operational Research 138(1), 332–348 (2002)

[7] Doumpos, M., Marinakis, Y., Marinaki, M., Zopounidis, C.: 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)

[8] Leroy, A., Mousseau, V., Pirlot, M.: Learning the parameters of a multiple criteria sorting method. In: Brafman, R., Roberts, F., Tsoukiàs, A. (eds.) Algorithmic Decision Theory, Lecture Notes in Computer Science, vol. 6992, pp. 219–233. Springer Berlin / Heidelberg (2011)

[9] Mousseau, V., Figueira, J., Naux, J.P.: Using assignment examples to infer weights for ELECTRE TRI method: Some experimental results. European Journal of Operational Research 130(1), 263–275 (2001)

[10] Mousseau, V., Slowinski, R.: Inferring an ELECTRE TRI model from assignment examples. Journal of Global Optimization 12(1), 157–174 (1998)

[11] Mousseau, V., Slowinski, R., Zielniewicz, P.: A user-oriented implementation of the ELECTRE TRI method integrating preference elicitation support. Computers & OR 27(7-8), 757–777 (2000)

[12] Ngo The, A., Mousseau, V.: Using assignment examples to infer category limits for the ELECTRE TRI method. Journal of Multi-criteria Decision Analysis 11(1), 29–43 (2002)

[13] Roy, B., Bouyssou, D.: Aide multicritère à la décision: méthodes et cas. Economica Paris (1993)

[14] Yu, W.: Aide multicritère à la décision dans le cadre de la problématique du tri: méthodes et applications. Ph.D. thesis, LAMSADE, Université Paris Dauphine, Paris (1992)

A piecewise linear approximation of PROMETHEE II's net flow scores

Stefan Eppe[1] **and Yves De Smet**

Abstract. PROMETHEE II is a prominent outranking method that builds a complete ranking on a set of actions by means of pairwise action comparisons. However, the number of comparisons increases quadratically with the number of actions, leading to computation times that may become prohibitive for large decision problems. Practitioners generally seem to alleviate this issue by down-sizing the problem, a solution that may not always be acceptable though. Therefore, as an alternative, we propose a piecewise linear model that approximates PROMETHEE II's net flow scores without requiring costly pairwise comparisons: our model reduces the computational complexity (with respect to the number of actions) from quadratic to linear, at the cost of some mis-ranked actions. Experimental results on artificial problem instances show a decreasing proportion of those mis-ranked actions as the problem size increases. This observation leads us to provide empirical bounds above which the PROMETHEE II-ranking of an action set is satisfyingly approximated by our piecewise linear model.

1 Introduction

Outranking methods represent one of the main families of multi-criteria decision aiding (MCDA) methods that are based on the pairwise comparison of potential actions [10]. However, despite the many applications reported in the literature [1, 5], outranking methods do suffer from a certain lack of scalability that is due mainly to computational limitations rather than to intrinsic flaws. This scalability issue arises in situations where large data sets must be handled, e.g., for geographical information analysis [8]; or where moderately large preference queries [9] have to be processed in parallel and quickly (for Internet product configuration interfaces). Indeed, providing the decision maker with an outranking-based evaluation on n actions comes at a cost of $O(n^2)$ pairwise action comparisons.

In this paper, we choose to focus on the PROMETHEE II method [2]. It is a representative and widely used outranking method [1] that builds a complete ranking on a set of actions by associating a so-called *net flow* score to each of them (Section 2). As already stated, evaluating these scores for larger decision problems becomes increasingly demanding in terms of execution time and memory usage. Although this difficulty tends to decrease as computational power of machines increases, the required time for a complete PROMETHEE II evaluation may still remain prohibitive for some particularly demanding applications.

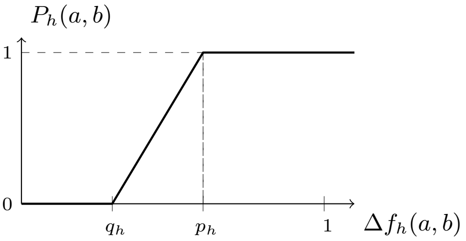


Figure 1. PROMETHEE's "V-shaped" preference function for criterion h, $a, b \in A$, where $\Delta f_h(a,b) = f_h(a) - f_h(b)$ in the case of a maximization problem. For each criterion, an indifference threshold q_h and a preference threshold p_h have to be provided by the decision maker.

Therefore, we propose an *ex ante* piecewise linear approximation of each action's net flow score (Section 2). After describing our experimental setup (Section 3), we empirically validate our model and show that even for relatively small action sets, our model is able to satisfyingly approximate the unicriterion net flow (Section 4). Along other observations, we also experimentally determine as from what instance size the ranking induced by a piecewise linear approximation is sufficiently close to the original PROMETHEE II-ranking.

2 A unicriterion piecewise linear approximation model

In this section, we first provide a brief description of the PROMETHEE II method. We then extend the original, discrete formulation to a continuous one, which will allow us to build a piecewise linear approximation of each action's net flow score. This second part represents the core of our contribution.

Let $A = \{a_1, \ldots, a_n\}$ be a set of n actions. Each action a_i, with $i \in I = \{1, \ldots, n\}$, is characterized by means of m evaluations $f_h(a_i)$, $\forall h \in H = \{1, \ldots, m\}$. To compare any pair of actions $a, b \in A$, we take, for each criterion, the "*V-shaped*" preference function, with indifference and preference thresholds that are respectively denoted q_h and p_h (Figure 1):

$$P_h(a,b) = \begin{cases} 0 & \text{, if } \Delta f_h(a,b) \leq q_h \\ \frac{\Delta f_h(a,b) - q_h}{p_h - q_h} & \text{, if } q_h < \Delta f_h(a,b) \leq p_h \\ 1 & \text{, if } p_h < \Delta f_h(a,b) \end{cases}$$

The pairwise action comparisons are aggregated for each action and provide the unicriterion net flow score

[1] Computer & Decision Engineering (CoDE) department, Polytechnic School of Brussels, Université Libre de Bruxelles, Belgium, email: stefan.eppe@ulb.ac.be

$\phi_h(a)$ on criterion h:

$$\phi_h(a) = \frac{1}{n-1} \sum_{b \in A} [\, P_h(a,b) - P_h(b,a) \,] \,.$$

Finally, the unicriterion scores are aggregated over all criteria through a weighted sum to yield that action's net flow score:

$$\phi(a) = \sum_{h \in H} w_h \phi_h(a), \tag{1}$$

where $w = \{w_1, \ldots, w_m\}$ is a vector of each criterion's relative importance, with $w_h > 0, \forall h \in H$, and $\sum_{h \in H} w_h = 1$. The interested reader may refer to [2] for a more detailed introduction to the PROMETHEE methods. Without loss of generality, we will consider a maximization problem and assume that evaluations lie in the interval $f_h(a_i) \in [0,1],\ \forall (i,h) \in I \times H$.

Our goal, in this paper, is to determine an approximation of an action's net flow score $\phi(a)$ that only depends on its evaluations $f_h(a)$, $\forall h \in H$, and on the preference parameters. It should not require any pairwise action comparison.

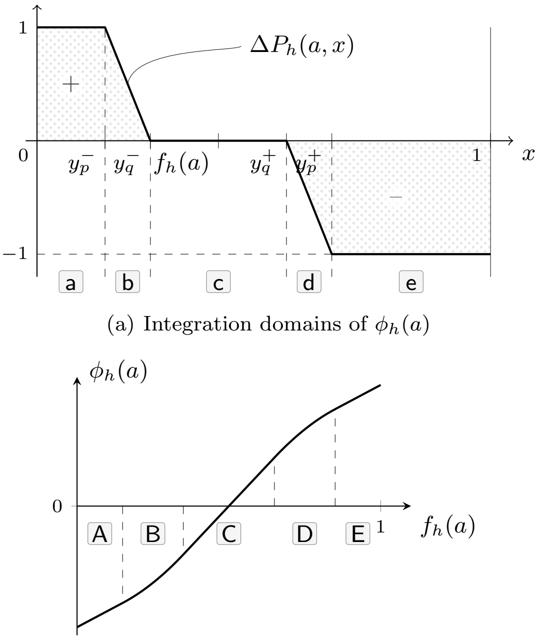
With this aim in mind, and given the mathematical form of (1), we start by focusing on the unicriterion terms of the weighted sum. Indeed, if we manage to determine an *ex ante* approximation of each unicriterion score $\phi_h(a)$, the global approximation would immediately be determined by (1).

For the sake of simplicity, we will assume that the actions are sorted in increasing order of their evaluation for the considered criterion: $f_h(a_i) \leq f_h(a_j)$, $\forall i < j$. We can thus rewrite the unicriterion net flow score of action a_i as

$$\phi_h(a_i) = \frac{1}{n-1} \left[\sum_{j=1}^{i-1} P(a_i, a_j) \; - \sum_{j=i+1}^{n} P(a_j, a_i) \right]$$

In this form, however, it is not easy to deduce any particular functional form. Therefore, we extend the formulation above to the case of an infinite set $\mathcal{A}$ of actions. Exploring the possible meanings of this continuous extension lays beyond the scope of this work. We will only consider it as a mathematical mean that could provide us some insight into the asymptotic behavior of unicriterion net flow scores. Still assuming that the actions are sorted in ascending order of their evaluations, we introduce the variable $x \in [0,1]$ which value identifies a considered action. We assume that the continuous distribution of actions along x is given by the density function $\rho(x)$, chosen such that $\int_0^1 \rho(\xi) d\xi = 1$. The unicriterion net flow can thus be rewritten as follows in the continuous case:

$$\phi_h^\infty(a) = \int_0^a P_h(a,\xi)\rho(\xi)d\xi - \int_a^1 P_h(\xi,a)\rho(\xi)d\xi$$



(a) Integration domains of $\phi_h(a)$

(b) Shape of the unicriterion net flow score

Figure 2. (a) The function $\Delta P_h(a,x) = P_h(a,x) - P_h(x,a)$ has to be integrated over the interval $x \in [0,1]$ in order to compute the unicriterion net flow's value $\phi_h(a) = \int_0^1 \Delta P_h(a,x)dx$. The shaded regions represent the positive and negative contributions to the unicriterion net flow's value $\phi_h(a)$, here for a coordinate $a = 0.4$. (b) Shape of the unicriterion net flow $\phi_h(a)$.

Let us consider this formulation for the particular case of a uniform distribution of actions: $\rho(\xi) = 1$, with, hence, $f_h(\xi) = \xi$. Introducing the following help variables (that depend on a):

$$\begin{cases} y_q^- &= \max\{\, 0 \,;\, a - q_h \,\} \\ y_p^- &= \max\{\, 0 \,;\, a - p_h \,\} \\ y_q^+ &= \min\{\, 1 \,;\, a + q_h \,\} \\ y_p^+ &= \min\{\, 1 \,;\, a + p_h \,\} \end{cases}$$

the integration range of (2) can be sliced into five segments, denoted $\boxed{a}$,…,$\boxed{e}$ (Figure 2(a)). Once integrated, we obtain the following formulation:

$$\begin{aligned} \phi_h^\infty(a) \;=\; & \underbrace{y_p^-}_{\boxed{a}} + \underbrace{\frac{(y_q^- - y_p^-)^2}{2(p_h - q_h)}}_{\boxed{b}} + \underbrace{0}_{\boxed{c}} \\ & - \underbrace{\frac{(y_p^+ - y_q^+)^2}{2(p_h - q_h)}}_{\boxed{d}} - \underbrace{\left(1 - y_p^+\right)}_{\boxed{e}} \end{aligned} \tag{2}$$

The unicriterion net flow $\phi_h^\infty(a)$ is composed of three terms: $\boxed{a}$, $\boxed{c}$, and $\boxed{e}$, of linearly increasing values, separated by two intervals: $\boxed{b}$ and $\boxed{d}$, with quadratic terms. The extent and contribution of each of them to the unicriterion net flow depends on both thresholds q_h and p_h, and also on action's a evaluation $f_h(a)$. In the general case, five different ranges can again be distinguished in ϕ_h^∞'s domain (Figure 2(b)):

$$\begin{array}{ll} & 0 = a \\ \boxed{\text{A}} & 0 < a < q_h \\ \boxed{\text{B}} & q_h \leq a < p_h \\ \boxed{\text{C}} & p_h \leq a < 1 - p_h \\ \boxed{\text{D}} & 1 - p_h \leq a < 1 - q_h \\ \boxed{\text{E}} & 1 - q_h \leq a < 1 \\ & 1 = a \end{array}$$

This general interpretation yields for the case where $p_h < \frac{1}{2}$. For higher values of p_h, the shape changes slightly, but can be determined analytically in a similar way.

Depending on the values of q_h and p_h, some ranges may be reduced to an empty range. For instance, if $q_h = p_h$, there are no quadratic terms and $\phi_h^\infty(a)$ reduces to a piecewise linear function. This form is particularly appealing for its simplicity and, as a further simplification, we approximate (2) by a piecewise linear function $\phi_h^{\text{PL}}(a)$ composed of three segments (whatever the threshold values) and defined by one unique parameter $\lambda_h = \frac{1}{2}(q_h + p_h)$:

$$\phi_h^{\text{PL}}(a) \quad = \quad \begin{cases} a + \lambda_h - 1 & \text{if } a < \lambda_h \\ 2a - 1 & \text{if } \lambda_h \leq a < 1 - \lambda_h \\ a - \lambda_h & \text{if } a > 1 - \lambda_h \end{cases} \qquad (3)$$

The arbitrary choice for this particular piecewise linear (PL) model is based on the following reasoning: As has been shown, the linear segments of (2) have characteristic slopes of $\frac{d\phi_h^\infty}{da} = 1$ for both "outer segments" ($\boxed{\text{a}}$ and $\boxed{\text{e}}$) and $\frac{d\phi_h^\infty}{da} = 2$ for the central segment ($\boxed{\text{c}}$). We want the PL-model to reflect this. Of course, we also want to keep the central symmetry around the coordinate $(\frac{1}{2}, 0)$. Finally, we impose that the extreme values $-\phi_h^\infty(0) = \phi_h^\infty(1) = 1 - \lambda_h$. As a resulting feature, the linear segments of our model intersect at symmetric coordinates $(\lambda_h, 2\lambda_h - 1)$ and $(1 - \lambda_h, 1 - 2\lambda_h)$.

Beyond its formal simplicity, it is noticeable that this function only has one single parameter, λ_h, that is the mean value of q_h and p_h. This remarkable property questions (at least for bigger problem instances) the practical advantage of requiring two parameters to determine a preference function. It also tends to show that the effects of indifference and preference parameters on an action's ranking do compensate each other in some way. This could shed a new light on the difficulty of eliciting these parameters [3]. Experiments that would only try to elicit the relative weight and parameter λ_h for each criterion could be run to verify this conjecture.

Finally, we aggregate the set of unicriterion approximations through a weighted sum, just like for the original mehtod, and we obtain an approximation of each action's net flow

$$\phi^{\text{PL}}(a) = \sum_{h=1}^{m} w_h \phi_h^{\text{PL}}(a), \qquad (4)$$

Algorithm 1: Standard experimental process that outputs a result vector κ of N_{trials} runs.

```
Input: $n$, $m$, $N_{\text{trials}}$
for $i = 1 \ldots N_{\text{trials}}$ do
    $A$ = randEvals($n, m$);
    $(w, q, p)$ = randPrefParams($m$);
    $R$ = computeNetFlowRanking($A, w, q, p$);
    $R^{PL}$ = computePLApproxRanking($A, w, q, p$);
    $\kappa_i$ = computeCRatio($R, R^{PL}$);
```

which induces a ranking over A. We denote $R^{\text{PL}}(a)$ the rank of action a based on our approximated model, and hope it to be as close as possible to action a's reference rank $R(a)$ obtained with the classical PROMETHEE II method.

3 Experimental setup

From an artificial continuous formulation, we have deduced a piecewise linear (PL) approximation that we hope to be applicable to finite action sets. We are now going to put our model to the test, comparing the rankings it generates with the reference ranking produced by the original PROMETHEE II method. Beyond the mere validation of our model, our main aim is to provide an empirical bound on the instance size above which PROMETHEE II's net flow scores are reasonably well approximated by our *ex ante* parametrized PL-function.

The experimental approach proposed in this paper consists (Algorithm 1) in generating a random instance of n actions over m criteria, as well as preference parameters (weights and thresholds) for each criterion. Therefrom, the rankings of the generated action set following respectively PROMETHEE II's original model (R) and our piecewise linear model (R^{PL}) are computed and their similarity compared. We use the resulting similarity measure to

1. validate the approach by showing that for reasonably sized instances, our PL-model satisfyingly approximates the PROMETHEE II ranking;
2. produce a table that provides an experimental numerical bound for the instance size, as from which the approximation quality reaches a required level.

To make things more concrete, we now provide some practical details about different aspects of the experimental setup:

Quality measure We define a *rank concordance ratio* κ, which is the ratio of the number of concordant action pairs, i.e., pairs that have the same relative rank order in both rankings, over the total number of pairwise action comparisons:

$$\kappa = \tfrac{1}{n(n-1)} \sum_{a,b \in A} c(a, b),$$

where

$$c(a,b) = \begin{cases} 1 & \text{, if } \left[\phi(a) \geq \phi(b) \wedge \phi^{PL}(a) > \phi^{PL}(b)\right] \\ & \quad \vee \left[\phi(a) > \phi(b) \wedge \phi^{PL}(a) \geq \phi^{PL}(b)\right] \\ 0 & \text{, otherwise} \end{cases}$$

indicates whether or not a rank difference between a pair of actions following respectively rankings R and R^{PL} is concordant. Although this measure is closely related to Kendall's τ rank correlation coefficient [6], we prefer the former because it allows taking possible ties into account.

Randomly generated instances We generate instances of n actions, evaluated on m criteria. For each generated instance, one type of distribution (Figure 3) is uniform randomly associated to each criterion.[2] The evaluations on each criterion are then randomly generated for all actions following that distribution. By doing so, we aim at producing results that are not (too strongly) biased by the features of one specific distribution. With this way of generating random instances, we implicitly assume that the evaluations are uncorrelated, which is an arguable hypothesis. Further tests with correlated criteria evaluations should be carried out in the future. Note also that the PL-model expressed by (2) assumes a uniform distribution. We will have to verify that mixed distributions do not affect the approximation's quality too much.

4 Results & discussion

Before delving into the empirical exploration of our model, we start this section by providing a first analysis of the statement (Section 2) that P2-rankings may depend on only one threshold-like parameter λ_h per criterion. When then proceed with several qualitative and quantitative investigations to validate the PL-model. Finally, we provide the results that we initially aimed for and that answer the following question: "*As from what instance size is it possible to satisfyingly approximate an action's net flow score by our piecewise linear model?*".

The compensating effect of Promethee II's threshold values

The piecewise linear approximation proposed in Section 2 only depends on one parameter, $\lambda_h = \frac{1}{2}(q_h + p_h)$. As already noted, this could suggest some sort of compensating effect between indifference and preference thresholds q_h and p_h in the Promethee II preference model. To verify this, we observe how the ranking of an action set A changes when the threshold values are altered (the weights remain unchanged). Practically, we take the ranking R_λ, induced by the threshold values $q_h = p_h = \lambda$, as a reference and compare it with the ranking R' induced by another pair of threshold values (q'_h, p'_h). The comparison is done through the rank concordance ratio $\kappa(R_\lambda, R')$. In particular, we investigate the case where $\lambda_h = 0.25$. At each run: 1) a random set of actions is generated as described in Section 3, as well as a random weight vector; 2) the concordance κ is computed between $R_{\lambda=0.25}$ and R', the latter being induced by threshold values q_h and p_h, where $p_h \in \{0, 0.005, \ldots, 0.500\}$ and $q_h \in \{0, 0.005, \ldots, p_h\}$. We finally compute $\kappa(R_\lambda, R')$'s 5% quantile for a series of 1000 runs, i.e., an approximation of the minimum concordance ratio reached with a probability of 95%.

The results (Figure 4) show, for all tested instance sizes, a symmetry with respect to the bisecting line $q_h + p_h = 2\lambda_h$. This tends to confirm the compensating role of q_h and p_h: the influence of their average value λ_h on Promethee II's ranking is higher than their individual values. As the instance size increases, the isolines become more and more parallel to this bisecting line. On the other hand, the similarity of induced rankings with the reference ranking R_λ decreases when the threshold pair tends to $(q_h, p_h) \to (0, 0.5)$. It is obviously the highest, i.e., $\kappa = 1$, for $q_h = p_h = \lambda$. The latter observation is particularly visible on smaller instances. The underlying reasons for this decrease should be investigated in a future paper.

Empirical validation of our model

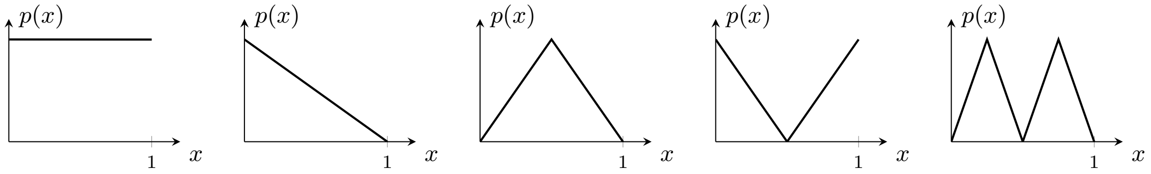
Table 1. Parameters used for the experimental investigation. Values in bold represent the most often used combinations provided in the results section.

Parameter		Value(s)
Number of actions	n	**5**,**10**,**50**,500,1000
Number of criteria	m	2,**3**,5,**7**,10
Ex post approximation models		LiR, **P3R**
Ex ante approximation models		**PLA**, PPA
Runs per instance config.	N_{trials}	1000

As a first validation, we visually compare both plots of Figure 5. It shows that, although the experimental results displayed in (b) are based on a relatively small instance of $n = 20$ randomly generated evaluations (with a uniform distribution), these results are close to the "theoretical" continuous results (a). This suggests that general features deduced from the theoretical model could also satisfyingly yield for practical instances. In a further step, we will investigate how the differences between the model and the practical results can be quantified.

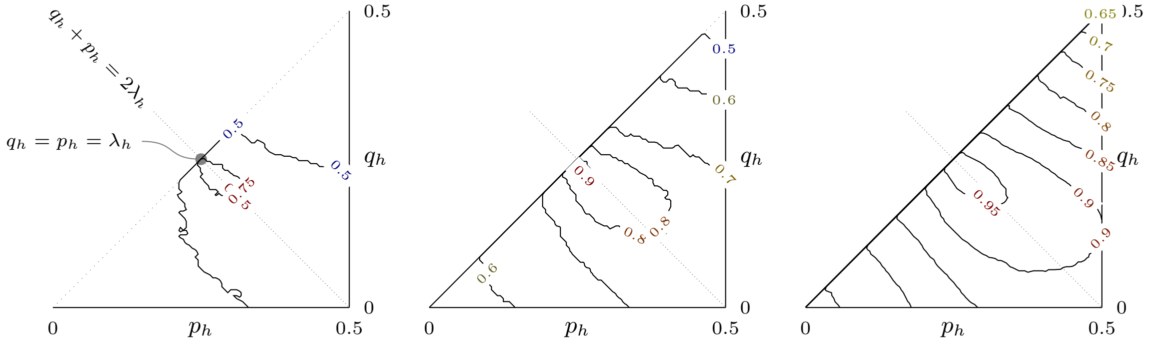
As a validation for our piecewise linear model, we compare the approximation quality, measured by the rank concordance ratio κ, with three other models: 1) *ex post* linear regression of P2-ranking; 2) *ex post* 3rd degree polynomial regression of P2-ranking (motivated by the shape of the net flow score); and 3) *ex ante* piecewise polynomial approximation model. The comparison (Figure 6) shows that:

[2] The attribution of a distribution function is independent for each criterion. Hence, the same distribution may be related to several criteria of the same instance.



(a) D1 (b) D2 (c) D3 (d) D4 (e) D5

Figure 3. Shape of the probability density functions $p(x)$ used to generate random instances.



(a) $n = 5$ actions (b) $n = 10$ actions (c) $n = 50$ actions

Figure 4. Isolines of 5%-quantile of rank concordance κ measured between the reference ranking R_λ corresponding to threshold values $q_h = p_h = \lambda_h$ and the ranking R' induced threshold given by the coordinate (q_h, p_h), with $p_h \in [0, 0.5]$ and $q_h \in [0, p_h]$. The statistics are computed on 1000 randomly generated instances (with "mixed distributions") of respectively 5, 10, and 50 actions, evaluated on 3 criteria. The threshold values are the same for all criteria.

1. the *ex post* 3rd degree polynomial regression offers the best approximation;
2. the PL-model and the piecewise polynomial approximation model give similar results that come close to the *ex post* results.

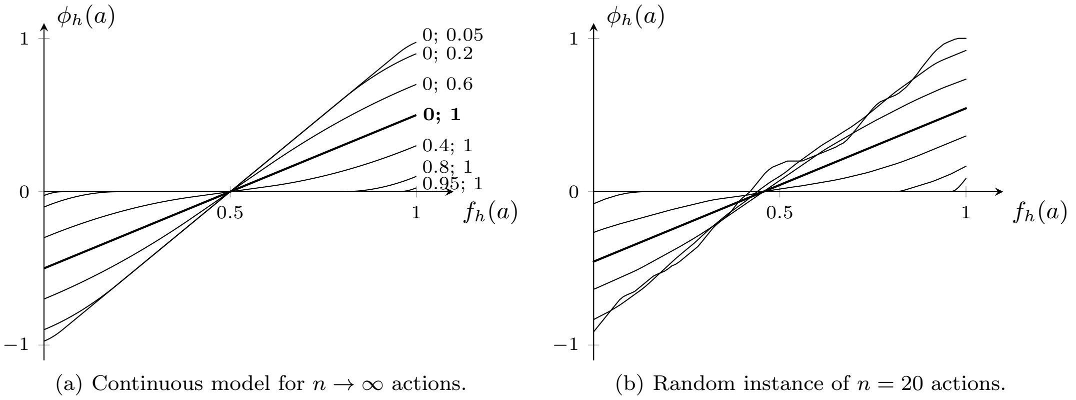
Empirical bounds for the use of our model

Figure 7 shows, for different instance sizes, the complement to 1 of the cumulated density function (CDF) of κ, for a series of 1000 runs. Concretely, the plots give the approximated probability of reaching at least a given similarity, measured by the rank concordance ratio κ. In the following, we will often refer to this ratio as a measure of quality: the higher κ, the better the approximation of PROMETHEE II's rankings (abbreviated by "P2-ranking" in the sequel) by our piecewise linear model. Several observations can be done on the basis of these plots:

1. While a higher number of actions increases the approximation's quality, changing the number of criteria has the opposite effect.
2. The quality curve converges to an "extreme curve" (approximated by the plots for $n = 1000$), which indicates that there exists an upper bound for the approximation quality. In other words, whatever the instance size, it will not in general be possible for our PL-model to induce the same ranking as PROMETHEE II.
3. Taking the opposite point of view, the results also show that a satisfying approximation (depending on a chosen quality level) can be reached, even for relatively small instance sizes that are frequently encountered in actual MCDA problems. *Example*: For instances of $n = 10$ actions and $m = 7$ criteria, a concordance of 90% can be reached with a probability of 87%. More numerical results are presented in Table 2.

Figure 8 shows the same results from another perspective. For a given probability P that measures some sort of required accuracy level of the approximation quality, the plots represent the minimum quality κ that is reached as a function of the number n of actions.

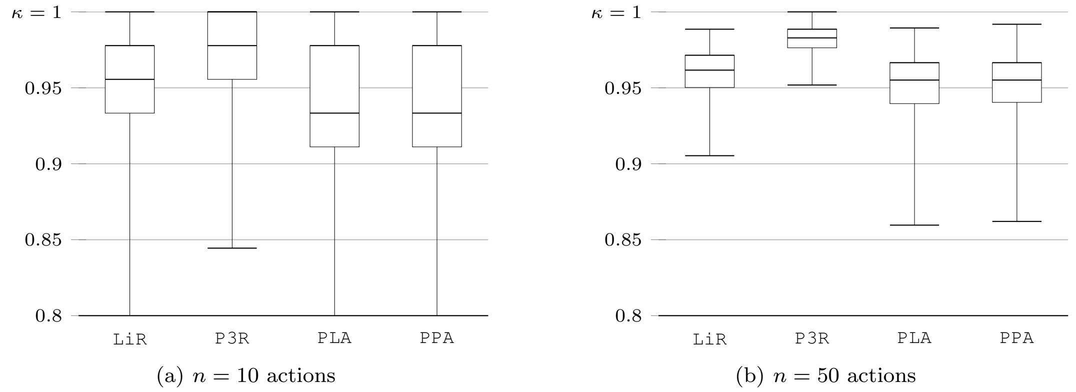
The question that naturally arises when using an approximation model, is to locate, if possible, regions where it is relatively better or worse. For our concern, we search for the actions that are not ranked appropriately with respect to the original P2-ranking. Figure 9(a) represents the distribution of mis-ranked actions with respect to the original rank. The result can be quantitatively explained by the shape of the plot that represents the average net flow score (over 1000 runs) for each rank position (Figure 9(b)). The



(a) Continuous model for $n \to \infty$ actions.

(b) Random instance of $n = 20$ actions.

Figure 5. The visual comparison of the theoretical continuous model (assuming $n \to \infty$) with a discrete instance of $n = 20$ actions tends to confirm the validity of the continuous approximation model. Indeed, the "theoretical" continuous functions (a) are very similar to the results obtained for a randomly generated discrete set of $n = 20$ actions (b). The pair of values attached to each plot of (a) are respectively the corresponding indifference and preference thresholds: "q_h; p_h". The same parameter values are used and appear in the same order for (b).



(a) $n = 10$ actions

(b) $n = 50$ actions

Figure 6. The box plots compare four types of net flow score approximation models: *ex post* linear regression (LiR); *ex post* 3-rd degree polynomial regression (P3R); *ex ante* piecewise linear approximation (PLA); and *ex ante* piecewise polynomial approximation (PPA). The results are shown for 1000 runs over mixed-distribution randomly generated action sets of 10 (resp. 50) actions and 7 criteria.

results show that the approximation should be even more satisfying than the rank concordance ratio indicates, since the actions ranked among the first or the last are often considered with more attention. We could think of an additional metric that takes this into consideration, by taking rank concordance of well and badly ranked actions more into consideration as the middle-ranked ones. This could, for instance be done by adapting the generalized Kendall's rank correlation τ [7] to our needs.

5 Conclusion

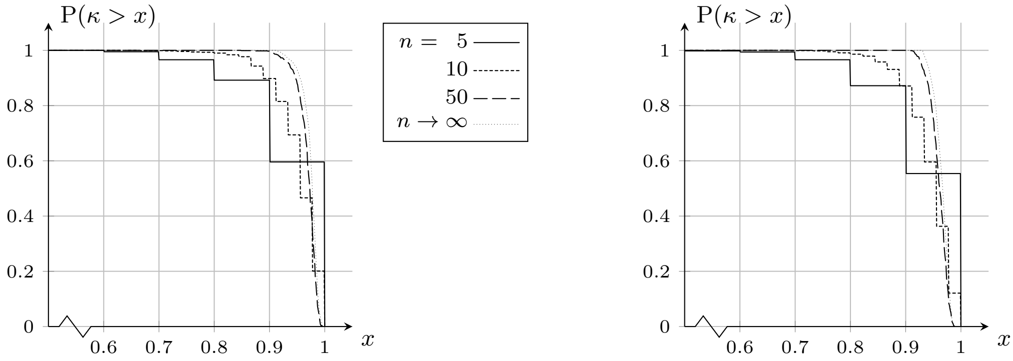
The PROMETHEE II method uses pairwise action comparisons to build a complete ranking over the set of considered alternatives. However, building this ranking is computationally demanding.This represents a significant drawback for PROMETHEE II when tackling MCDA ranking problems that are very large and/or have to be computed very often. Being able to "switch" to an approximated model with linear complexity when a compromise between ranking accuracy and computation speed is affordable would therefore be of great practical interest.

In this paper, we propose such an approximated model. It is based on a piecewise linear approximation of the PROMETHEE II net flow. Taking the usual PROMETHEE preference parameters, i.e., weights and indifference/preference thresholds, an approximation of an action's net flow is provided by a function that only depends on its evaluations. The approximated net flows are then used to determine a complete ranking over the set of considered actions.

Under the assumption of criteria independence, an experimental study has provided us with quantitative evaluations of the approximation's quality, yielding the minimum quality level that can be reached with a given

Table 2. Probability $P(\kappa > x)$ to reach a rank concordance ratio κ that is at least as high than x for mixed-distribution randomly generated instances of different sizes.

	$n = 5$					$n = 10$					$n = 50$				
$x \setminus m$	2	3	5	7	10	2	3	5	7	10	2	3	5	7	10
0.70	0.98	0.97	0.97	0.97	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
0.80	0.93	0.89	0.89	0.87	0.89	0.99	0.99	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00
0.90	0.93	0.89	0.89	0.87	0.89	0.92	0.90	0.88	0.87	0.86	0.99	1.00	1.00	1.00	1.00
0.95	0.67	0.60	0.59	0.55	0.55	0.75	0.69	0.61	0.60	0.58	0.89	0.88	0.73	0.75	0.69
0.99	0.67	0.60	0.60	0.55	0.55	0.24	0.20	0.13	0.12	0.13	0.07	0.04	0.00	0.00	0.00



(a) $m = 3$ criteria

(b) $m = 7$ criteria

Figure 7. The probability $P(\kappa > x)$ to reach a rank concordance ratio κ that is at least as high than x for mixed-distribution randomly generated instances evolves as the number n of actions and m of criteria increases.

probability. This has been done for a variety of instance sizes (from 5 to 1000 actions and from 2 to 7 criteria). Practically, these results provide indicative bounds on the instance size as from which the ranking may be approximated by our model, showing that already moderately large instances (10 to 20 actions) could be approximated with an "acceptable" loss of ranking accuracy. At this stage, the presented results should be considered as first results that have to be deepened on a wider range of random instances, in particular integrating correlated multivariate evaluations.

On a more theoretical level, the piecewise linear formulation of our model only depends on the threshold's average value, $\lambda_h = \frac{1}{2}(q_h + p_h)$, for each criterion. This suggests that the ranking may, to a certain extent, only depend on that unique parameter, instead of a set of two threshold parameters. This feature, that has been partially confirmed by a set of experiments, provides a deeper insight into the PROMETHEE II preference model internal structure. Both the theoretical relation between threshold parameters as well as its practical implications (e.g. for preference eliciting procedures) should be studied in more depth in the future.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable feed-back. Although not all remarks could be integrated in the length they would have deserved, they will contribute to improve further developments. Furthermore, Stefan Eppe acknowledges support from the META-X Arc project, funded by the Scientific Research Directorate of the French Community of Belgium.

REFERENCES

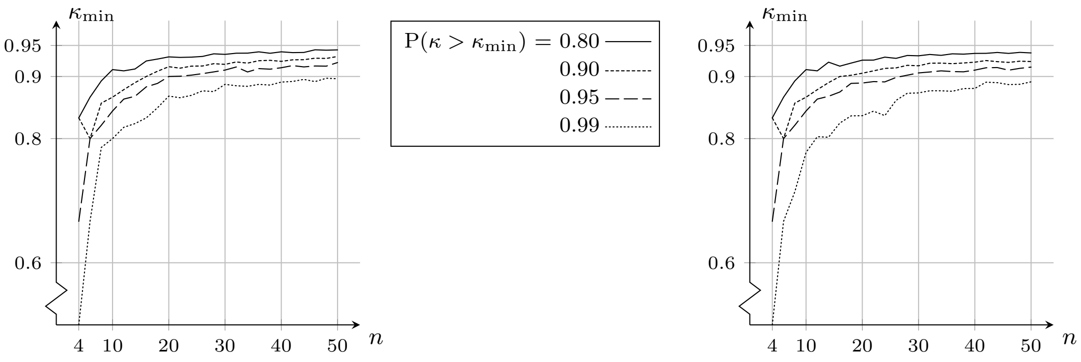
[1] Majid Behzadian, R.B. Kazemzadeh, A. Albadvi, and M. Aghdasi. PROMETHEE: A comprehensive literature review on methodologies and applications. *European Journal of Operational Research*, 200(1):198–215, 2010.

[2] Jean-Pierre Brans and Bertrand Mareschal. PROMETHEE methods. In Figueira et al. [4], chapter 5, pages 163–195.

[3] Stefan Eppe, Yves De Smet, and Thomas Stützle. A bi-objective optimization model to eliciting decision maker's preferences for the PROMETHEE II method. In Ronen I. Brafman, Fred S. Roberts, and Alexis Tsoukiàs, editors, *Algorithmic Decision Theory, Third International Conference, ADT 2011*, volume 6992 of *Lecture Notes in Computer Science*, pages 56–66. Springer, Heidelberg, Germany, 2011.

[4] José Rui Figueira, Salvatore Greco, and Matthias Ehrgott, editors. *Multiple Criteria Decision Analysis, State of the Art Surveys*. Springer, 2005.

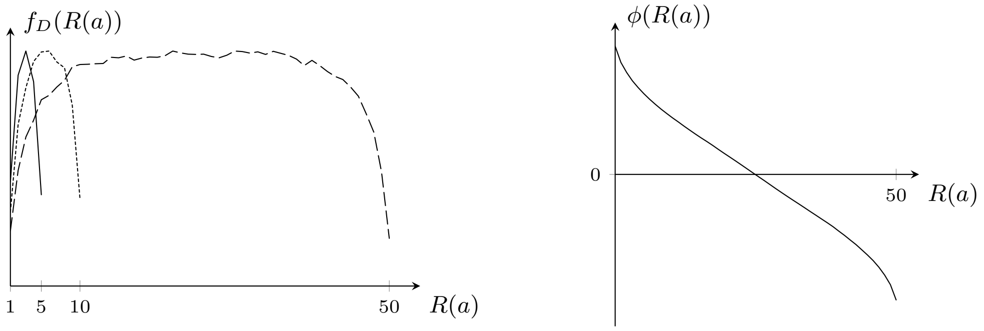
[5] José Rui Figueira, Vincent Mousseau, and



(a) $m = 3$ criteria

(b) $m = 7$ criteria

Figure 8. For different probabilities $\mathrm{P}(\kappa > \kappa_{\min})$, the plots show the minimum quality $\kappa_{\min}$ that can be achieved with respect to the number of actions n and depending also on the number m of criteria.



(a) Distribution f_D of misranked actions as a function of the rank $R(a)$

(b) Shape of the net flow score function ϕ as a function of the rank $R(a)$

Figure 9. The distribution of mis-ranked actions with the PL-model with respect to the original P2-ranking displayed in (a) shows, for different instance sizes, that extreme ranks, i.e., the best few as well as the worst few, are relatively stable when compared to the centrally-ranked actions. This is due to the average shape of the net flow score plotted with respect to the rank position (b): $\phi(R(a))$ is more discriminant in the extreme rank regions. Results are plotted for 1000 repetitions on randomly generated instances (with mixed distributions) of 5, 10, and 50 actions with 3 criteria.

Bernard Roy. ELECTRE methods. In Figueira et al. [4], chapter 4, pages 133–162.

[6] Maurice Kendall and Jean D. Gibbons. *Rank Correlation Methods.* Edward Arnold, 1990.

[7] Ravi Kumar and Sergei Vassilvitskii. Generalized distances between rankings. In Michael Rappa, Paul Jones, Juliana Freire, and Soumen Chakrabarti, editors, *Proceedings of the 19th International Conference on World Wide Web, WWW 2010.* ACM press, New York, NY, 2010.

[8] Oswald Marinoni. A discussion on the computational limitations of outranking methods for land-use suitability assessment. *International Journal of Geographical Information Science*, 20(1):69–87, 2006.

[9] Olivier Pivert and Grégory Smits. Towards an efficient processing of outranking-based preference queries. In Salvatore Greco, Bernadette Bouchon-Meunier, Giulianella Coletti, Mario Fedrizzi, Benedetto Matarazzo, and Ronald R. Yager, editors, *Information Processing and Management of Uncertainty, 14th International Conference, IPMU2012*, volume 300 of *Communications in Computer and Information Science*, pages 471–480. Berlin / Heidelberg, 2012.

[10] Bernard Roy. Paradigms and challenges. In Figueira et al. [4], chapter 1, pages 3–24.

Session 3

Invited speaker : Paolo Viappianì

CNRS-LIP6, Université Pierre et Marie Curie, Paris
"***Principled Techniques for Utility-based Preference Elicitation in Conversational Systems***",

Preference elicitation is an important component of many applications, such as decision support systems and recommender systems. It is however a challenging task for a number of reasons. First, elicitation of user preferences is usually expensive (w.r.t. time, cognitive effort, etc.). Second, many decision problems have large outcome or decision spaces. Third, users are inherently "noisy" and inconsistent.
Adaptive utility elicitation tackles these challenge by representing the system knowledge about the user in form of "beliefs" about the possible utility functions, that are updated following user responses ; elicitation queries can be chosen adaptively given the current belief. In this way, one can often make good (or even optimal) recommendations with sparse knowledge of the user's utility function.
We analyze the connection between the problem of generating optimal recommendation sets and the problem of generating optimal choice queries, considering both Bayesian and regret-based elicitation. Our results show that, somewhat surprisingly, under very general circumstances, the optimal recommendation set coincides with the optimal query.

Session 4

- "*Using Choquet integral in Machine Learning : what can MCDA bring ?*",
  D. Bouyssou[1], M. Couceiro[1], C. Labreuche[2], J.-L. Marichal[3] and B. Mayag[1]
  [1] CNRS-Lamsade, Université Paris Dauphine,
  [2] Thales,
  [3] Université du Luxembourg.

- "*On the expressiveness of the additive value function and the Choquet integral models*",
  P. Meyer[1] and M. Pirlot[2]
  [1] Institut Télécom, Télécom Bretagne,
  [2] MATHRO, Faculté Polytechnique, UMONS

- "*Using set functions for multiple classifiers combination*",
  F. Rico[1], A. Rolland[1],
  [1] Laboratoire ERIC - Université Lumière Lyon

- "*Preference Learning using the Choquet Integral*",
  E. Hüllermeier[1]
  [1] Department of Mathematics and Computer Science, Philipps-Universität Marburg, Germany

Using Choquet integral in Machine Learning: what can MCDA bring?

D. Bouyssou[1] M. Couceiro[1] C. Labreuche[2] J.-L. Marichal[3] B. Mayag[1]

Abstract. In this paper we discuss the Choquet integral model in the realm of Preference Learning, and point out advantages of learning simultaneously partial utility functions and capacities rather than sequentially, i.e., first utility functions and then capacities or vice-versa. Moreover, we present possible interpretations of the Choquet integral model in Preference Learning based on Shapley values and interaction indices.

1 Introduction

The first application of the Choquet integral in computer science appeared in the late 80's in the field of decision under uncertainty [21], and early 90's in the fields of multi-criteria decision making (MCDM) [7] and data mining [6, 25]. Recently, it has also been used in machine learning (ML) [24]) and preference learning (PL) [4]. The use of the Choquet integral in MCDM and data mining for almost 20 years has lead to a wide literature dealing with both theoretical (axiomatizations) and practical (methodologies, algorithms) aspects [10]. The new fields of ML and PL can benefit from this huge literature. We focus on two aspects in this paper.

The first aspect concerns partial utility functions. As an aggregation function of n input variables, the Choquet integral requires that these variables are *commensurate.* By commensurate, we mean that a same value (say 0.5) taken by two different input variables must have the same meaning. In MCDA, this meaning refers to the degree of satisfaction to criteria. For instance, value 0.5 corresponds to half-satisfaction. The reason why the Choquet requires commensability is that it compares the values taken by the n variables. This commensurability property is obtained by introducing partial utility functions over the attributes. The use of partial utility functions is well-established in MCDA. They are much less used in ML and PL. Sometimes, the attributes are aggregated without having being normalized. When attributes are normalized, the partial utility functions are fixed a priori and not learnt. The main point of the paper is to show that, fixing utility functions a priori significantly reduces the expressivity of the model. For instance, if we only consider three criteria, when the utility functions are fixed, it is not difficult to find two comparisons that cannot be represented by a Choquet integral model: in fact 2 comparisons are sufficient (see Section 3.1). Now, when utility functions are not fixed, the simplest example we came up with that is not representable by a Choquet integral and partial utility functions is composed of 6 comparisons (see Section 4.1) with only two attributes. Using conditional relative importance, we also give a non-representable example composed of 11 comparisons with three attributes (see Section 4.2). Back to the case of two attributes, we show in Section 4.3, some sufficient conditions under which the preference relation can be represented by a Choquet integral and partial utility functions.

The second aspect is on the interpretation of the model. Murofushi proposed to use the Shapley value as an importance index [18], and later introduced an interaction index [19]. These two concepts are often used to interpret a capacity. The use of these indices might be debatable as one may argue that the user is interested in the interpretation of the Choquet integral and not the capacity. We recall some results – apparently not known from the community in ML and PL – showning that the Choquet integral can be interpreted in terms of Shapley and interaction indices. These results show that the Shapley value is actually equal to the mean value of the discrete derivative of the Choquet integral over all possible vectors in $[0,1]^n$. This assumes that the set of possible alternatives is uniformly distributed in $[0,1]^n$. We show in Section 5 how to extend these results to non uniform distributions which arises often in ML or even in MCDA. Some connections with the definition of the Shapley value and interaction indices on non-Boolean lattices are given.

2 Preliminaries

2.1 The Choquet integral

Let us denote by $N = \{1, \ldots, n\}$ a finite set of n criteria and $X = X_1 \times \cdots \times X_n$ the set of actions (also called alternatives or options), where for each $i \in N$, X_i represents the set of possible levels on criterion i. We refer to function $u_i : X_i \to \mathbb{R}$, $i = 1, \ldots, n$, as utility function.

The Choquet integral [9, 10, 17, 16] is based on a *capacity* μ defined as a set function from the powerset of criteria 2^N to $[0,1]$ such that:

1. $\mu(\emptyset) = 0$
2. $\mu(N) = 1$
3. $\forall A, B \in 2^N,\ [A \subseteq B \Rightarrow \mu(A) \leq \mu(B)]$ (monotonicity).

For an alternative $x := (x_1, ..., x_n) \in X$, the expression of the Choquet integral w.r.t. a capacity μ is given by:

$$C_\mu((u_1(x_1), \ldots, u_n(x_n))) \quad := \quad \sum_{i=1}^{n}(u_{\tau(i)}(x_{\tau(i)}) \quad - u_{\tau(i-1)}(x_{\tau(i-1)}))\ \mu(\{\tau(i), \ldots, \tau(n)\})$$

where τ is a permutation on N such that $u_{\tau(1)}(x_{\tau(1)}) \leq u_{\tau(2)}(x_{\tau(2)}) \leq \cdots \leq u_{\tau(n-1)}(x_{\tau(n-1)}) \leq u_{\tau(n)}(x_{\tau(n)})$, and $u_{\tau(0)}(x_{\tau(0)}) := 0$.

The preferential information of the decision maker is represented by a binary relation $\succsim$ over X where $\succ$ is the asymmetric part of $\succsim$.

Let $\Pi(2^N)$ be the set of permutations on N, and $Z_\tau = \{z \in [0,1]^n\ :\ z_{\tau(1)} \geq \cdots \geq z_{\tau(n)}\}$, for $\tau \in \Pi(2^N)$. The Choquet integral $C_\mu(x)$ is clearly a weighted sum in each domain Z_τ. The

[1] LAMSADE, University Paris Dauphine, Place du Maréchal de Lattre de Tassigny, 75116 Paris, France. Email: {denis.bouyssou, miguel.couceiro, brice.mayag}@dauphine.fr

[2] Thales Research & Technology, 1 avenue Augustin Fresnel, 91767 Palaiseau Cedex- France. Email: christophe.labreuche@thalesgroup.com

[3] Faculté des Sciences, de la Technologie et de la Communication, 6, rue Richard Coudenhove-Kalergi, L-1359 Luxembourg. Email: jean-luc.marichal@uni.lu

weights of criteria change from a domain Z_τ to another one $Z_{\tau'}$, for $\tau, \tau' \in \Pi(2^N)$. Two alternatives are called *comonotone* if they belong to a same set Z_τ. The Choquet integral is additive for all comonotone alternatives [20].

2.2 Interpretation of a capacity

A capacity is a complex object (it contains 2^n parameters), hence it is useful to provide an interpretation of μ.

The Shapley value [22] is often used in MCDA as a tool to interpret a capacity [19, 7, 8]. Actually, the concept of Shapley value comes from cooperative game theory and has been axiomatized in this framework [26]. The Shapley value describes how the worth obtained by all players shall be fairly redistributed among themselves [27].

Let us give a construction of the Shapley value in the spirit of cost allocation (cooperative game theory). N is interpreted here as the set of players and $\mu(S)$ is the cost of the cheapest way to serve all agents in S, ignoring the players in $N \setminus S$ altogether. All players of N agree to participate in the collective use of the common technology or public goods. Consider an ordering $\tau \in \Pi(2^N)$ of the players. Assume that the players are served in the order given by this permutation. Once the k first players have been served, the marginal cost of serving the next player according to the permutation is $\mu(\{\tau(1), \ldots, \tau(k+1)\}) - \mu(\{\tau(1), \ldots, \tau(k)\}) =: h^\tau_{\tau(k+1)}(\mu)$. The Shapley value allocates to agent i her expected marginal cost over all possible orderings of agents [22]:

$$\begin{aligned}\phi_i(\mu) &:= \frac{1}{n!} \sum_{\tau \in \Pi(2^N)} h_i^\tau(\mu) \\ &= \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n-|S|-1)!}{n!} \Delta_i \mu(S)\end{aligned}$$

where $\Delta_i\mu(S) := \mu(S \cup \{i\}) - \mu(S)$. Coefficient $\frac{|S|!(n-|S|-1)!}{n!}$ is the probability that coalition S corresponds precisely to the set of players preceding player i in a giving ordering.

The *interaction index* [19] between criteria i and j is defined by

$$I_{ij}(\mu) := \sum_{A \subset N \setminus \{i,j\}} \frac{|A|!(n-|A|-2)!}{(n-1)!} \Delta_{i,j}\mu(A)$$

where $\Delta_{i,j}\mu(A) := \mu(A \cup \{i,j\}) - \mu(A \cup \{i\}) - \mu(A \cup \{j\}) + \mu(A)$. A positive (resp. negative) interaction depicts a positive (resp. negative) synergy between criteria – both criteria need to be satisfied (resp. it is sufficient that only one criterion is met).

This interaction index was extended to any coalition A of criteria [5]:

$$I_A(\mu) = \sum_{B \subseteq N \setminus A} \frac{(n-|B|-|A|)!|B|!}{(n-|A|+1)!} \Delta_A \mu(B),$$

where $\Delta_A\mu(B) = \sum_{K \subseteq A} (-1)^{|A \setminus K|} \mu(B \cup K)$. In particular we have

$$I_{\{i\}}(\mu) = \phi_i(\mu) \quad \text{and} \quad I_{\{i,j\}}(\mu) = I_{i,j}(\mu).$$

3 Choquet integral: the importance of learning utility functions and capacities simultaneously

3.1 The limitation of Choquet integral: a classical example

A classical example that shows the limitation of the Choquet integral model is [9]:

The students of a faculty are evaluated on three subjects Mathematics (M), Statistics (S) and Language skills (L). All marks are taken from the same scale from 0 to 20. The evaluations of eight students are given by the table below:

	1 : Mathematics (M)	2 : Statistics (S)	3 : Language (L)
A	16	13	7
B	16	11	9
C	6	13	7
D	6	11	9
E	14	16	7
F	14	15	8
G	9	16	7
H	9	15	8

To select the best students, the dean of the faculty expresses his preferences:

- for a student good in Mathematics, Language is more important than Statistics

$$\Longrightarrow A \prec B \quad \text{and} \quad E \prec F,$$

- for a student bad in Mathematics, Statistics is more important than Language

$$\Longrightarrow D \prec C \quad \text{and} \quad H \prec G.$$

The two preferences $A \prec B$ and $D \prec C$ lead to a contradiction with the arithmetic mean model because

$$\begin{cases} A \prec B \Rightarrow 16\ w_M + 13\ w_S + 7\ w_L < 16\ w_M + 11\ w_S + 9\ w_L \\ \\ D \prec C \Rightarrow 6\ w_M + 11\ w_S + 9\ w_L < 6\ w_M + 13\ w_S + 7\ w_L. \end{cases}$$

Furthermore it is not difficult to see that the other two preferences, $E \prec F$ and $H \prec G$, are not representable by a Choquet integral C_μ since

$$\begin{cases} E \prec F \Rightarrow 7 + 7\mu(\{M,S\}) + 2\mu(\{S\}) < 8 + 6\mu(\{M,S\}) + \mu(\{S\}) \\ \\ H \prec G \Rightarrow 8 + \mu(\{M,S\}) + 6\mu(\{S\}) < 7 + 2\mu(\{M,S\}) + 7\mu(\{S\}) \end{cases}$$

$$i.e. \begin{cases} E \prec F \Rightarrow \mu(\{M,S\}) + \mu(\{S\}) < 1 \\ \\ H \prec G \Rightarrow \mu(\{M,S\}) + \mu(\{S\}) > 1 \end{cases}$$

An important remark in this example is that we try to find a capacity by assuming that the utility functions are fixed. If the latter are not fixed, then $E \prec F$ and $H \prec G$ can be modeled by C_μ, for instance, using these following utility functions:

	1 : Mathematics (M)	2 : Statistics (S)	3 : Language (L)
E	$u_M(14) = 16$	$u_S(16) = 16$	$u_L(7) = 7$
F	$u_M(14) = 16$	$u_S(15) = 15$	$u_L(8) = 8$
G	$u_M(9) = 9$	$u_S(16) = 16$	$u_L(7) = 7$
H	$u_M(9) = 9$	$u_S(15) = 15$	$u_L(8) = 8$

Indeed these utility functions lead to the system

$$\begin{cases} E \prec F \Rightarrow 2\mu(\{M,S\}) - \mu(\{M\}) < 1 \\ \\ H \prec G \Rightarrow \mu(\{M,S\}) + \mu(\{S\}) > 1 \end{cases}$$

Hence a capacity μ such that $\mu(\{M,S\}) = \mu(\{M\}) = \mu(\{S\}) = 0.6$ can be found. The utility function given above show that for the DM, the interpretation of "a good mark" in mathematics and "a good mark" in statistics is different. Such an interpretation is not in contradiction with the definition of commensurate scales: for $x_i \in X_i$ and $x_j \in X_j$,

$u_i(x_i) \geq u_j(x_j)$ iff the DM considers x_i at least as good as x_j.

Of course, if we assume that $u_M(a) = u_S(a) = u_L(a)$, for all $a \in [0,20]$, then $E \prec F$ and $H \prec G$ remain not representable by C_μ. This is not surprising because such situations can be viewed as the representation of preferences in decision under uncertainty where the Choquet integral model is well characterized [20, 21]. The four alternatives E, F, G, H are comonotone and thus the preferences $E \prec F$ and $H \prec G$ violate comonotone additivity.

To show the limitation of the Choquet integral in MCDA, we look for an example where the utility functions are not fixed a priori. This is the purpose of the next section.

4 Example non representable by a Choquet integral

We wish to known under which condition $\succsim$ is representable by a Choquet integral, i.e. there exists n utility functions $u_i : X_i \to \mathbb{R}$ and a capacity μ such that for all $x, y \in X$

$$x \succsim y \Longrightarrow C_\mu(u_1(x_1), \ldots, u_n(x_n)) \geq C_\mu(u_1(y_1), \ldots, u_n(y_n)). \tag{1}$$

4.1 A counter-example with 2 criteria

Let $\succsim$ be a weak order on the set $X = X_1 \times X_2$. We are interested in conditions that would guarantee that $\succsim$ can be represented using a Choquet integral model, i.e., that there is a real valued function u_1 on X_1 a real valued function u_2 on X_2 and positive real numbers λ_1, ω_1, such that:

$$x \succsim y \iff V(x) \geq V(y),$$

where V is a real valued function on X such that:

$$V(x) = \begin{cases} \lambda_1 u_1(x_1) + (1-\lambda_1)u_2(x_2) & \text{if } u_1(x_1) \geq u_2(x_2), \\ \omega_1 u_1(x_1) + (1-\omega_1)u_2(x_2) & \text{otherwise.} \end{cases}$$

Such a model is clearly a particular case of the model studied in [2] in which

$$V(x) = F(u_1(x_1), u_2(x_2)), \tag{2}$$

F being nondecreasing in its two arguments. We suppose that the conditions underlying the latter model hold. They are given in [2]. We now give an example of a weak order on X that cannot be represented using a Choquet integral model whatever the capacity and the functions u_1 and u_2.

Example 1 *Let* $X_1 = \{a_1, b_1, c_1, d_1, e_1, f_1\}$ *and* $X_2 = \{a_2, b_2, c_2, d_2, e_2, f_2\}$.

Suppose that the relation $\succsim$ *is such that:*

$$\begin{array}{c} (a_1, e_2) \sim (b_1, d_2) \\ (c_1, d_2) \sim (a_1, f_2) \\ (c_1, e_2) \not\sim (b_1, f_2) \end{array} \tag{3}$$

and

$$\begin{array}{c} (d_1, b_2) \sim (e_1, a_2) \\ (f_1, a_2) \sim (d_1, c_2) \\ (f_1, b_2) \not\sim (e_1, c_2) \end{array} \tag{4}$$

It is easy to find a weak order on X that satisfies the conditions in [2] and that includes the relations (3) *and* (4). *Moreover, it is not difficult to choose this weak order in such a way as to satisfy* (2) *together with:*

$$\begin{array}{c} u_1(a_1) \leq u_1(b_1) \leq u_1(c_1) \leq u_1(d_1) \leq u_1(e_1) \leq u_1(f_1) \\ u_2(a_2) \leq u_2(b_2) \leq u_2(c_2) \leq u_2(d_2) \leq u_2(e_2) \leq u_2(f_2) \end{array} \tag{5}$$

Since each of triple of relations (3) *and* (4) *violates the Thomsen condition [13], they cannot be represented using an additive model.*

Considering the first triple, this implies that it is impossible that we have

$$\begin{array}{c} u_1(a_1) \geq u_2(f_2), \\ \textit{or} \\ u_2(d_2) \geq u_1(c_1). \end{array}$$

Indeed, if it were the case the representation for the elements in the triple would be additive, so that the Thomsen condition would be satisfied.

Similarly, considering the second triple, it is impossible that we have

$$\begin{array}{c} u_1(d_1) \geq u_2(c_2) \\ \textit{or} \\ u_2(a_2) \geq u_1(f_1). \end{array}$$

Hence, we must have:

$$\begin{array}{c} u_1(a_1) < u_2(f_2), \\ \textit{and} \\ u_2(d_2) < u_1(c_1), \\ \textit{and} \\ u_1(d_1) < u_2(c_2) \\ \textit{and} \\ u_2(a_2) < u_1(f_1). \end{array}$$

It is not difficult to see that, together with (5) *this leads to contradiction.*

Hence any weak order on X that has a representation in model (2) *satisfying* (5) *and that contains the relations in* (3) *and* (4) *cannot be represented using the Choquet integral model.*

4.2 A counter-example with 3 criteria

Let $n = 3$. We assume that there is an order on attribute 1. For instance, X_1 is a real interval and the utility function is increasing (e.g. X_1 represents the elements of a revenue). Another example: $X_1 = \{$ "very bad", "bad", "medium", "good", "very good", $\}$, where "very bad" is worse than " bad", etc. The ordering on X_1 is denoted by $\leq$ and the strict ordering by $<$.

We choose two elements on attributes 2 and 3:

$$y_2, z_2 \in X_2 \text{ and } y_3, z_3 \in X_3.$$

We choose now eleven elements on attribute 1:

$$x_1^1, x_1^2, \ldots, x_1^{11} \in X_1 \text{ with } x_1^1 < x_1^2 < \cdots < x_1^{11}.$$

We assume that the decision maker provides the following preferential information:

$$\begin{aligned}
&(x_1^1, y_2, y_3) \succ (x_1^1, z_2, z_3)\\
&(x_1^2, y_2, y_3) \prec (x_1^2, z_2, z_3)\\
&(x_1^3, y_2, y_3) \succ (x_1^3, z_2, z_3)\\
&(x_1^4, y_2, y_3) \prec (x_1^4, z_2, z_3)\\
&(x_1^5, y_2, y_3) \succ (x_1^5, z_2, z_3)\\
&(x_1^6, y_2, y_3) \prec (x_1^6, z_2, z_3)\\
&(x_1^7, y_2, y_3) \succ (x_1^7, z_2, z_3)\\
&(x_1^8, y_2, y_3) \prec (x_1^8, z_2, z_3)\\
&(x_1^9, y_2, y_3) \succ (x_1^9, z_2, z_3)\\
&(x_1^{10}, y_2, y_3) \prec (x_1^{10}, z_2, z_3)\\
&(x_1^{11}, y_2, y_3) \succ (x_1^{11}, z_2, z_3)
\end{aligned}$$

This idea of this example is to introduce sufficiently many comparisons such that there necessarily exist three comparisons of comonotonic alternatives leading to a contradiction.

Lemma 1 *The previous example is not representable by (1).*

Proof: Assume for a contradiction that there exist utility functions u_1, u_2, u_3 and a capacity μ representing the previous example. Note that $u_1(x_1^1) < u_1(x_1^2) < \cdots < u_1(x_1^{11})$. Let $V = \{u_2(y_2), u_2(z_2), u_3(y_3), u_3(z_3)\}$. These four elements split the real line $\mathbb{R}$ into at most five intervals $(-\infty, v_1]$, $[v_1, v_2]$, $[v_2, v_3]$, $[v_3, v_4]$ and $[v_4, +\infty)$, where $v_1 \leq v_2 \leq v_3 \leq v_4$ and $V = \{v_1, v_2, v_2, v_4\}$.

It is not difficult to see that among $u_1(x_1^1), u_1(x_1^2), \ldots, u_1(x_1^{11})$, at least three of them necessarily belong to the same interval. These three values necessarily correspond to three successive elements, denoted by x_1^k, x_1^{k+1} and x_1^{k+2}. Hence, in the preferential information, the comparison obtained from x_1^k and x_1^{k+2} are the same and are opposite to the comparison obtained with x_1^{k+1}. More precisely, we have two cases:

- In the first case, we have

$$\begin{aligned}
&(x_1^k, y_2, y_3) \succ (x_1^k, z_2, z_3)\ ,\\
&(x_1^{k+1}, y_2, y_3) \prec (x_1^{k+1}, z_2, z_3)\\
&\text{and}\\
&(x_1^{k+2}, y_2, y_3) \succ (x_1^{k+2}, z_2, z_3).
\end{aligned} \tag{6}$$

As $u_1(x_1^k)$, $u_1(x_1^{k+1})$ and $u_1(x_1^{k+2})$ belong to the same interval, the vectors $(u_1(x_1^k), u_2(y_2), u_3(y_3))$, $(u_1(x_1^{k+1}), u_2(y_2), u_3(y_3))$ and $(u_1(x_1^{k+2}), u_2(y_2), u_3(y_3))$ are comonotone, and $(u_1(x_1^k), u_2(z_2), u_3(z_3))$, $(u_1(x_1^{k+1}), u_2(z_2), u_3(z_3))$ and $(u_1(x_1^{k+2}), u_2(z_2), u_3(z_3))$ are comonotone. The Choquet integral is a weighted sum for all comonotone vectors. We denote by (w_1^y, w_2^y, w_3^y) the weights of criteria (obtained from the capacity μ) for the comonotone vectors $(u_1(x_1^k), u_2(y_2), u_3(y_3))$, $(u_1(x_1^{k+1}), u_2(y_2), u_3(y_3))$ and $(u_1(x_1^{k+2}), u_2(y_2), u_3(y_3))$. We denote by (w_1^z, w_2^z, w_3^z) the weights of criteria (obtained from the capacity μ) for the comonotone vectors $(u_1(x_1^k), u_2(z_2), u_3(z_3))$, $(u_1(x_1^{k+1}), u_2(z_2), u_3(z_3))$ and $(u_1(x_1^{k+2}), u_2(z_2), u_3(z_3))$. Hence (6) gives

$$\begin{aligned}
&u_1(x_1^k)\, w_1^y + u_2(y_2)\, w_2^y + u_3(y_3)\, w_3^y\\
&\quad > u_1(x_1^k)\, w_1^z + u_2(z_2)\, w_2^z + u_3(z_3)\, w_3^z
\end{aligned} \tag{7}$$

$$\begin{aligned}
&u_1(x_1^{k+1})\, w_1^y + u_2(y_2)\, w_2^y + u_3(y_3)\, w_3^y\\
&\quad < u_1(x_1^{k+1})\, w_1^z + u_2(z_2)\, w_2^z + u_3(z_3)\, w_3^z
\end{aligned} \tag{8}$$

$$\begin{aligned}
&u_1(x_1^{k+2})\, w_1^y + u_2(y_2)\, w_2^y + u_3(y_3)\, w_3^y\\
&\quad > u_1(x_1^{k+2})\, w_1^z + u_2(z_2)\, w_2^z + u_3(z_3)\, w_3^z
\end{aligned} \tag{9}$$

Combining (7) with (8) gives

$$(u_1(x_1^k) - u_1(x_1^{k+1}))\, (w_1^y - w_1^z) > 0$$

As $u_1(x_1^k) < u_1(x_1^{k+1})$, we obtain $w_1^y < w_1^z$. Combining (9) with (8) gives

$$(u_1(x_1^{k+2}) - u_1(x_1^{k+1}))\, (w_1^y - w_1^z) > 0$$

As $u_1(x_1^{k+2}) > u_1(x_1^{k+1})$, we obtain the opposite inequality $w_1^y > w_1^z$. Hence a contradiction is attained.

- In the second case, we have

$$\begin{aligned}
&(x_1^k, y_2, y_3) \prec (x_1^k, z_2, z_3)\ ,\\
&(x_1^{k+1}, y_2, y_3) \succ (x_1^{k+1}, z_2, z_3) \text{ and}\\
&(x_1^{k+2}, y_2, y_3) \prec (x_1^{k+2}, z_2, z_3).
\end{aligned} \tag{10}$$

We proceed similarly and a contradiction is also raised.

■

From the two counter-examples presented above, we can deduce some necessary conditions to represent a preference by the Choquet integral model when utility functions and capacity are unknown a priori. Therefore we hope to entirely characterized this model in the future works. The search of this characterization led us to obtain a first sufficient condition in the case of two criteria.

4.3 Sufficient conditions for representability by Choquet integrals with 2 criteria

Let X_1, X_2 be two arbitrary chains (linearly ordered sets), and let $\succsim$ be a partial relation on $X_1 \times X_2$, extendable to a (total) preference relation on $X_1 \times X_2$ (i.e., which does not violate reflexivity, transitivity and the Pareto condition), and let $\succ$ be its nonsymmetric part ... Denote by $D(\succsim)$ the universe of R, i.e., the set of elements $x \in X_1 \times X_2$ that appear in some couple in $\succsim$.

Proposition 1 *Every partial relation $\succsim$ on $X_1 \times X_2$ for which $D(\succsim)$ is a finite antichain (w.r.t. the componentwise ordering of $X_1 \times X_2$) can be extended to a (total) preference relation on $X_1 \times X_2$ that is representable by a Choquet integral.*

Proof: Suppose that $D(\succsim) = \{x_1, \dots, x_n\}$, $x_i = (x_{i1}, x_{i2})$, is an antichain. Without loss of generality, we assume that $x_{11} < \dots < x_{n1}$ and $x_{12} > \dots > x_{n2}$; the other possible case can be dealt with similarly. We shall construct utility functions $u_t \colon X_t \to \mathbb{R}$, $t = 1, 2$, and a capacity $\mu \colon 2^N \to [0,1]$ such that $x_i \succsim x_j$ implies $C_\mu(u_1(x_{i1}), u_2(x_{i2})) \leq C_\mu(u_1(x_{j1}), u_2(x_{j2}))$.

Since $\succsim$ is extendable to a (total) preference relation on $X_1 \times X_2$, we can partition $D(\succsim)$ into (indifference) classes $C_0, C_1, \dots$ that are defined recursively as follows:

1. C_0 contains all maximal elements for $\succsim$, i.e., elements $y \in D(\succsim)$ such that there is no $z \in D(\succsim)$ for which $y \succ z$;
2. if $C_0, C_1, \dots, C_K$ have been defined, then C_{K+1} contains all $y \in D(\succsim)$ such that $y \succ z$ for some $z \in C_K$, and there is no $z' \in D(\succsim) \setminus \bigcup_{0 \leq t \leq K} C_t$ such that $y \succ z'$.

Let $C_0, C_1, \dots, C_T$ be the thus defined classes.

Consider the capacity $\mu \colon 2^N \to [0,1]$ given by $\mu(\{1\}) = \mu(\{2\}) = \frac{1}{3}$ and $\mu(\{1,2\}) = 1$. Hence, $C_\mu(u_1(a_1), u_2(a_2)) = \frac{2}{3}u_1(a_1) + \frac{1}{3}u_2(a_2)$ if $u_1(a_1) \leq u_2(a_2)$; otherwise, $C_\mu(u_1(a_1), u_2(a_2)) = \frac{2}{3}u_2(a_2) + \frac{1}{3}u_1(a_1)$.

We construct $u_t \colon X_t \to \mathbb{R}$ on $\{x_{1t}, \dots, x_{nt}\}$, $t = 1, 2$, as follows. Let $s := \min\{k : x_k \in C_0\}$. Set $u_1(x_{s1}) = u_2(x_{s2}) = n$. Hence $C_\mu(u_1(x_{j1}), u_2(x_{j2})) = n$.

Also, note that $u_1(x_{k1}) \leq u_2(x_{k2})$ if $k < s$, and $u_1(x_{k1}) \geq u_2(x_{k2})$ if $k > s$.

Now, take a sufficiently small $\epsilon > 0$, say $\epsilon = \frac{1}{n}$. For each $1 \leq i \leq n$ such that $x_i \in C_K$, define

1. $u_1(x_{i1}) = \big(n - (\frac{|j-i|}{2}) - K\epsilon\big)$ and $u_2(x_{i2}) = \big(n + |j-i| + K\epsilon\big)$ if $k < s$, and
2. $u_1(x_{i1}) = \big(n + |j-i| + K\epsilon\big)$ and $u_2(x_{i2}) = \big(n - (\frac{|j-i|}{4}) - K\epsilon\big)$, otherwise.

It is not difficult to verify that for every $0 \leq S \leq T$ and $y = (y_1, y_2) \in C_S$, we have $C_\mu(u_1(y_1), u_2(y_2)) = n - \frac{S\epsilon}{3}$, and the proof is now complete. ■

5 Interpretation of the Choquet integral model

5.1 The $[0,1]^n$ case

In the context of MCDA, the Shapley value can be seen as the mean importance of criteria and is thus a useful tool to interpret a capacity [7, 8]. The interpretation of Section 2.2 of the Shapley value is not satisfactory in MCDA since it completely ignores the use of the Choquet integral.

The interpretation of the Shapley value (and the Shapley interaction indices) for the Choquet integral is basically due to J.L. Marichal who noticed that (see [15, proposition 5.3.3 page 141] and also [11, Definition 10.41 and Proposition 10.43 page 369])

$$I_S(\mu) = \int_{[0,1]^n} \Delta_S C_\mu(z) dz$$

where, for any function f, $\Delta_S f$ is defined recursively by

$$\Delta_S f(z) = \Delta_i(\Delta_{S \setminus \{i\}} f)(x) \quad \text{for any } i \in S$$
$$\Delta_i f(z) = f(z|z_i = 1) - f(z|z_i = 0)$$

The Shapley value appears as the mean of relative amplitude of the range of C_μ w.r.t. criterion i, when the remaining variables take random values. What is true with Shapley value is also true for interaction indices.

The following lemma is not difficult to prove:

Lemma 2 *We have*

$$I_S(\mu) = \int_{[0,1]^n} \frac{\partial^{|S|} C_\mu}{\partial z_S}(z)\, dz$$

where the partial derivative is piecewise continuous.

Here the partial derivative is the local importance of C_μ at point z.

5.2 The case of a subset of $[0,1]^n$

The set of options that the decision maker finds feasible is often far from covering the whole space X. The following example shows that only a subset denoted by Ω of X may be realistic.

Example 2 (Situation awareness) *Consider a surveillance system that generates alerts from the information provided by several sensors such as cameras and radars. The system provides a situation awareness of the environment, gathering the identification of the intruder and its accurate localization [23].*

We are interested in assessing the quality of information provided by the system. To this end, we access the difference between what the system displays to the user and the real situation. Three criteria are considered.

- Relevance of identity information*: this is the difference between the identity that is obtained by the system and the real identity of the intruder. The determination of a wrong identity has strong consequences on the level of threat associated to the intruder.*
- Rough localization*: There are several particular assets that must be protected in the area that is covered by the surveillance system. Three areas of interest are defined around the assets: the alert zone which is the area at close range of the assets, the warning zone which is the area at medium range of the assets, and the rest of the area. There is a procedure which indicates the action that must be performed by an operator when an intruder is in one of these three zones. The system identifies the area to which intruders belong. Clearly, the identification of a wrong area has a critical consequence on the safety (if an intruder at close range is not seen as being in that area) or the relevance (false alarms) of the system.*

- Fine localization*: This is the accuracy of the intruders localization made by the system, i.e., the distance between the localization given by the system and the real one. The decision maker needs to know the accurate location of intruders, especially, in the alert area in order to perform a dissuasive action on the intruders.*

The last two criteria quantify the consequence with respect to two different points of view attached to localization. These two criteria are statistically correlated. Indeed, when the second criteria is not met, which entails a crude error, then it is not possible that the last criteria is well-satisfied. This implies that the satisfaction of last criterion cannot be better than that of the second criterion. Hence $x \in X$ such that $x_3 > x_2$ is not feasible.

In MCDA, our starting point is the subset Ω of realistic options in $[0,1]^n$. One can assume that Ω is convex and has a non-zero measure, as it is the case in Example 2. For $\tau \in \Pi(2^N)$ and $z \in Z_\tau$, those coalitions used in the computation of the Choquet integral w.r.t. z are $\emptyset, \{\tau(1)\}, \{\tau(1),\tau(2)\}, \ldots, \{\tau(1),\ldots,\tau(n)\}$. The following set

$$\mathcal{T} := \{\tau \in \Pi(2^N)\ ,\ \mathrm{Int}(X_\tau)\cap\Omega \neq \emptyset\}$$

contains those permutations that are reached when computing the Choquet integral of the elements of Ω. The set of coalitions that are used in the previous computations is then

$$\mathcal{F} = D(\mathcal{T}) := \bigcup_{\tau\in\mathcal{T}} \{\emptyset, \{\tau(1)\}, \{\tau(1),\tau(2)\}, \\ \ldots, \{\tau(1),\ldots,\tau(n)\}\}.$$

A *convex geometry* on N is a family $\mathcal{F}$ of subsets of N satisfying the following properties [3, 12]

(i) $\emptyset \in \mathcal{F}$, $N \in \mathcal{F}$,
(ii) $S \in \mathcal{F}$ and $T \in \mathcal{F}$ implies that $S \cap T \in \mathcal{F}$,
(iii) $\forall S \in \mathcal{F}\setminus\{N\}$, $\exists i \in N \setminus S$ such that $S \cup \{i\} \in \mathcal{F}$.

Note that properties *(i)* and *(ii)* allow to define a closure operator $\overline{S} = \bigcap\{T \in \mathcal{F}\ :\ S \subseteq T\}$ for every $S \subseteq N$. The set of *extreme points* of a subset $S \in \mathcal{F}$ is defined as $\mathrm{ext}(S) = \{i \in S\ :\ S\setminus\{i\} \in \mathcal{F}\}$. From this set, one can define a shelling process. Starting with the whole set N, one may successively eliminate extreme points until the empty set is obtained. This process defines maximal chains of $\mathcal{F}$. A *chain in $\mathcal{F}$ from $S \in \mathcal{F}$ to $T \in \mathcal{F}$*, with $S \subset T$, is a set of nested elements of $\mathcal{F}$ of the form $S = K_{i_1} \subset K_{i_2} \subset \cdots \subset K_{i_{m-1}} \subset K_{i_m} = T$ with $|K_{i_l}| = i_l$ for $l = 1,\ldots,m$, and $i_1 < i_2 < \cdots < i_m$. A *maximal chain of $\mathcal{F}$ from $S \in \mathcal{F}$ to $T \in \mathcal{F}$*, with $S \subset T$, is a chain of $\mathcal{F}$ from S to T for which $m = t-s+1$ (i.e. $i_l = s+l-1$ for all l) with the previous notation. A *maximal chain of $\mathcal{F}$* is a maximal chain of $\mathcal{F}$ from $\emptyset$ to N.

Another interesting case is when the points x are not uniformly spread over $[0,1]^n$. A particular case is when there are some values in $[0,1]^n$ that are infeasible. Then, when computing the Choquet integral, some permutations may never occur, and thus some terms $\mu(S)$ (for some coalitions S) may never be used. The Shapley value has been defined for the situation of "forbidden" coalitions.

Let $\mathcal{F}$ be a convex geometry defined on N. A capacity on $\mathcal{F}$ is a function $\mu : \mathcal{F} \to \mathbb{R}$ satisfying the boundary and monotonicity conditions. Bilbao [1] defined the Shapley value of μ as follows

$$\phi_i^{\mathcal{F}}(\mu) = \sum_{S\subseteq N\setminus\{i\}\,:\,S,S\cup\{i\}\in\mathcal{F}} \frac{C_{\mathcal{F}}(S,S\cup\{i\})}{C_{\mathcal{F}}}\left[\mu(S\cup\{i\}) - \mu(S)\right], \tag{11}$$

where $C_{\mathcal{F}}$ is the total number of maximal chains of $\mathcal{F}$ and $C_{\mathcal{F}}(S, S\cup\{i\})$ is the number of maximal chains of $\mathcal{F}$ going through S and $S \cup \{i\}$.

A *reduced game* describes the situation where the players in a coalition P never play separately. As a consequence, they can be identified to a unique player denoted by $[P]$. Let $N_{[P]} := (N\backslash P)\cup\{[P]\}$. Let $\eta_P : \mathcal{P}_P(N) \to 2^{N_{[P]}}$ be defined by $\eta_P(S) = S$ if $P \not\subseteq S$ and $\eta_P(S) = (S\setminus P)\cup\{[P]\}$ otherwise. Given $\mathcal{F}$, the definition of the set of allowed coalitions $\mathcal{F}^{N_{[P]}}$ on $N_{[P]}$ is as follows [14]

$$\mathcal{F}^{N_{[P]}} = \eta_P(\mathcal{T}_P) = \{\eta_P(S)\ :\ S \in \mathcal{T}_P\}\,,$$

where $\mathcal{T}_P$ is the set of the elements of all chains $\emptyset = S_0 \subset \cdots \subset S_k \subset S_{k+p} \subset \cdots \subset S_n = N$ of elements of $\mathcal{F}$ with $|S_i| = i$, $S_{k+p} = S_k \cup P$ and $S_k \in \mathcal{M}_P$, and where $\mathcal{M}_P := \{S \subseteq N\setminus P\ :\ \forall T \subseteq P\ ,\ S\cup T \in \mathcal{F}\}$. The interaction index $I_P^{\mathcal{F}}(\mu)$ of coalition P w.r.t. a capacity μ has the expression [14]

$$I_P^{\mathcal{F}}(\mu) = \sum_{S\in\mathcal{M}_P} \frac{C_{\mathcal{F}^{N_{[P]}}}(S, S\cup\{[P]\})}{C_{\mathcal{F}^{N_{[P]}}}}\Delta_P\mu(S). \tag{12}$$

Lemma 2 can be extended to the current setting in the following way:

Lemma 3 *We have*

$$I_S^{\mathcal{F}}(\mu) = \int_{\cup_{\tau\in\mathcal{T}}Z_\tau} \frac{\partial^{|S|}C_\mu}{\partial z_S}(z)\,dz$$

where the partial derivative is piecewise continuous.

This formula clearly shows that $I_S^{\mathcal{F}}(\mu)$ is interpreted as the interaction among criteria S for the Choquet integral. Expression (12) provides a combinatorial formulae to compute $I_S^{\mathcal{F}}(\mu)$.

Note that $\cup_{\tau\in\mathcal{T}}Z_\tau$ appears as an approximation of the feasibility domain Ω. When this approximation is not so good, it is possible to compute the interaction index by the following expression

$$\int_\Omega \frac{\partial^{|S|}C_\mu}{\partial z_S}(z)\,dz.$$

This computation might be complex when Ω is itself complex.

6 Conclusion

We have shown the gain in terms of expressivity that is obtained when the partial utility functions are constructed at the same time as the Choquet integral. With only two attributes, an example of non representativity is constructed. It is very special in the sense that the alternatives take special values on a grid. Moreover, again with two attributes, when the learning examples use only alternatives that belong to an antichain, then we have shown that any weak order over these alternatives, that does not violates Pareto condition and transitivity, is representable by a Choquet integral and partial utility functions. Clearly this result is wrong when the partial utility functions are a priori fixed. With three criteria, using the idea of conditional relative importance, we need 11 learning examples to contradicts the Choquet integral model. We believe that these examples are important to construct axiomatic characterizations of the Choquet integral and its utility functions.

Next, the Shapley index and interaction indices often used to interpret a capacity can also be used to interpret a Choquet integral. Actually the interaction index among criteria S is the integral over $[0,1]^n$ of the partial derivative of the Choquet integral w.r.t. criteria in S. This can be easily extended to the cases when the set of feasible alternatives is not $[0,1]^n$ but a subset. The corresponding Shapley and interaction indices are then extension of the original indices on convex geometries.

We hope we have convinced the community working on ML and PL on the importance of learning not only the capacity but also partial utility functions.

REFERENCES

[1] J. M. Bilbao. Axioms for the Shapley value on convex geometries. *European Journal of Operational Research*, 110:368–376, 1998.

[2] D. Bouyssou and M. Pirlot. Preferences for multiattributed alternatives: Traces, dominance, and numerical representations. *Journal of Mathematical Psychology*, 48(3):167–185, 2004.

[3] P.H. Edelman and R.E. Jamison. The theory of convex geometries. *Geom. Dedicata*, 19:247–270, 1985.

[4] J. Fürnkranz and E. Hüllermeier. *Preference Learning.* Springer, 2011.

[5] M. Grabisch. k-order additive discrete fuzzy measures and their representation.

[6] M. Grabisch. A new algorithm for identifying fuzzy measures and its application to pattern recognition. In *Int. Joint Conf. of the 4th IEEE Int. Conf. on Fuzzy Systems and the 2nd Int. Fuzzy Engineering Symposium*, pages 145–150, Yokohama, Japan, March 1995.

[7] M. Grabisch. The application of fuzzy integrals in multicriteria decision making. *European J. of Operational Research*, 89:445–456, 1996.

[8] M. Grabisch. Alternative representations of discrete fuzzy measures for decision making. *Int. J. of Uncertainty, Fuzziness, and Knowledge Based Systems*, 5:587–607, 1997.

[9] M. Grabisch and C. Labreuche. Fuzzy measures and integrals in MCDA. In J. Figueira, S. Greco, and M. Ehrgott, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, pages 565–608. Springer, 2005.

[10] M. Grabisch and Ch. Labreuche. A decade of application of the Choquet and Sugeno integrals in multi-criteria decision aid. *4OR*, 6:1–44, 2008.

[11] M. Grabisch, J.L. Marichal, R. Mesiar, and E. Pap. *Aggregation functions.* Cambridge University Press, 2009.

[12] B. Korte, L. Lovász, and R. Schrader. *Greedoids.* Springer-Verlag, Berlin, 1991.

[13] D. H. Krantz, R. D. Luce, P. Suppes, and A. Tversky. *Foundations of measurement*, volume 1: Additive and polynomial representations. Academic Press, New York, 1971.

[14] Ch. Labreuche. Interaction indices for games on combinatorial structures with forbidden coalitions. *European Journal of Operational Research*, 214:99–108, 2011.

[15] J.-L. Marichal. *Aggregation operators for multicriteria decision aid.* PhD thesis, University of Liège, 1998.

[16] B. Mayag, M. Grabisch, and C. Labreuche. A characterization of the 2-additive Choquet integral through cardinal information. *Fuzzy Sets and Systems*, 184(1):84–105, 2011.

[17] B. Mayag, M. Grabisch, and C. Labreuche. A representation of preferences by the Choquet integral with respect to a 2-additive capacity. *Theory and Decision*, 71(3):297–324, 2011.

[18] T. Murofushi. A technique for reading fuzzy measures (I): the Shapley value with respect to a fuzzy measure. In *2nd Fuzzy Workshop*, pages 39–48, Nagaoka, Japan, October 1992. In Japanese.

[19] T. Murofushi and S. Soneda. Techniques for reading fuzzy measures (III): interaction index. In *9th Fuzzy System Symposium*, pages 693–696, Sapporo, Japan, May 1993.

[20] D. Schmeidler. Integral representation without additivity. *Proc. of the Amer. Math. Soc.*, 97(2):255–261, 1986.

[21] D. Schmeidler. Subjective probability and expected utility without additivity. *Econometrica*, 57(3):571–587, 1989.

[22] L. S. Shapley. A value for n-person games. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games, Vol. II*, number 28 in Annals of Mathematics Studies, pages 307–317. Princeton University Press, 1953.

[23] A.N. Steinberg and C.L. Bowman. Revisions to the JDL data fusion model. In D. Hall and L. Llinas, editors, *Handbook of multisensor data fusion - Chapter 2.* CRC Press, Boca Raton, FL, 2001.

[24] A. Fallah Tehrani, W. Cheng, K. Dembczyn'ski, and E. Hüllermeier. Learning monotone nonlinear models using the choquet integral. *Machine Learning*, 89(1-2):183–211, 2012.

[25] A. Verkeyn, D. Botteldooren, and B. De Baets. Genetic learning of fuzzy integrals accumulating human-reported environmental stress. *Applied Soft Computing*, 11:305–314, 2011.

[26] R. J. Weber. Probabilistic values for games. In A. E. Roth, editor, *The Shapley Value. Essays in Honor of Lloyd S. Shapley*, pages 101–119. Cambridge University Press, 1988.

[27] H. P. Young. Cost allocation. In *Fair allocation. Proceedings of the Symposia in Applied Mathematics*, pages 69–94, Providence, RI, 1985.

On the expressiveness of the additive value function and the Choquet integral models.

Patrick Meyer
Institut Télécom, Télécom Bretagne
UMR CNRS 6285 Lab-STICC
Technopôle Brest Iroise CS 83818
F-29238 Brest Cedex 3, France
Université européenne de Bretagne, France
patrick.meyer@telecom-bretagne.eu

Marc Pirlot
UMONS Université de Mons
Faculté Polytechnique
9 rue de Houdain, B-7000 Mons, Belgium
marc.pirlot@umons.ac.be

Abstract

Recent - and less recent - work has been devoted to learning additive value functions or a Choquet capacity to represent the preference of a decision maker on a set of alternatives described by their performance on the relevant attributes. In this work we compare the ability of related models to represent rankings of such alternatives. Our experiments are designed as follows. We generate a number of alternatives by drawing at random a vector of evaluations for each of them. We then draw a random order on these alternatives and we examine whether this order is representable by a simple weighted sum, a Choquet integral with respect to a 2- or 3-additive capacity, an additive value function in general or a piecewise-linear additive value function with 2 or 3 pieces. We also generate non preferentially independent data in order to test to which extent 2- or 3-additive Choquet integrals allow to represent the given orders. The results explore how representability depends on varying the numbers of alternatives and criteria.

Key words : Preference representation, additive value function, Choquet integral, weighted sum

1 Introduction

Additive value functions occupy a dominant position in the models used for representing the preferences of a decision maker (DM) on alternatives described by their performance on several attributes [7], [1]. This model implies preferential independence [7, page 32], which however does not mean that it can be used for representing any preference relation that satisfies this condition.

A simple example to show that this independence condition can easily be violated is that of a shopkeeper who wishes to rent a new showroom. The showrooms are evaluated according to their surface, their price and the city area in which they are located. The shopkeeper considers that in a commercially attractive area he prefers the showroom which is expensive and large over the cheap and small one, whereas in a commercially unattractive area he prefers the cheap and small showroom over the expensive and large one. This situation does not satisfy the preference independence condition and cannot be represented by an additive value function.

In the late 1980's another model for representing preferences in the multi-attribute context has emerged. The use of the Choquet integral – which is better known in the context of decision under risk [10] - is advocated in the multi-attribute case when a form ofi nteraction between criteria is presumed [2].

A simple example is that of the selection process of students applying for graduate studies in management (Grabisch and Labreuche, 2004). Students are evaluated by their past performance in mathematics, statistics and language skills. Since skills in mathematics and statistics are correlated (mathematics and statistics are, to some extent, redundant attributes), for students good at mathematics, the jury responsible for the selection prefers a student with good linguistic skills to one who is good at statistics. Things will go the other way around for students that have a weakness in mathematics. In such a situation, there is a (negative) interaction between the criteria mathematics and statistics and a violation of preference independence.

Note however that interaction between criteria is a property that should not be identified with the fact that the preference independence hypothesis is violated: preferences that can be represented by a Choquet integral do not necessarily satisfy the preference independence condition, but they do satisfy weaker forms ofi ndependence such as comonotonic independence [11, page 111] and weak separability [1]. Note that the property of comonotonic independence not easy to define in this context as it presupposes commensurability of the attributes' scales.

The goal of the present work is to study the "expressiveness" of these aggregation models, i.e. their ability to represent preferences. In particular, we wish to give answers to the following questions:

1. What is the expressiveness of the Choquet integral models compared to that of the general additive value and the weighted sum models ?

2. What is the expressiveness of the general additive value model compared to that of piecewise linear additive value ones ?

3. How do the Choquet integral models behave when confronted to non preferentially independent data ?

In this study, we assume that the Choquet integral is computed directly on the vector of performances of the alternatives, without any recoding of these performances by marginal value functions. The additive value function model, on the contrary, takes the preferences of the DM on each attribute into account by modelling them using possibly non linear functions defined on the range of values that is relevant for each attribute. In this setting, there obviously are examples in which the preference can be represented by means of an additive value function but not using a Choquet integral. One may also imagine a model in which the Choquet integral is applied to a vector of marginal value functions. Such a model would be more expressive since it would encompass the additive value function model. We briefly explain in the conclusion how it is possible to get an indication on how much more expressive such a model is as compared to the simple additive value function model involving the same marginal value functions.

In order to assess the relative expressiveness of these models, series of computational experiments are performed which are described in Section 2. We then analyze the results, draw some conclusions and suggest further investigations that seem interesting. Note finally that this article extends our work from [8] by more exhaustive experiments and by answering further questions on the expressiveness of these models.

2 Experimental setting

We denote by $(a_1, \ldots, a_m)$ the performance vector associated with an alternative a. Without loss of generality, we suppose here that the alternatives performances are assessed by values in the $[0, 1]$ interval. Let us present the various models which are tested in this work:

1. The weighted sumfor which the score of alternative a is computed as $\sum_{i=1}^{m} w_i a_i$, where w_i are weights that need to be determined.

2. The additive value function for which the score of alternative a is computed as $\sum_{i=1}^{m} u_i(a_i)$ where u_i are (marginal value) functions from the $[0, 1]$ interval into itself that need be determined. Two special cases of the general model are also considered, in which u_i is a piecewise linear marginal value function. In the first variant, u_i has two linear pieces corresponding to a division of the $[0, 1]$ interval into two equal parts. In the second variant there are three linear pieces, the $[0, 1]$ interval being divided in three equal sub-intervals.

3. A Choquet integral in which the score of alternative a is computed using 2-additive and 3-additive capacities. In the case of a 2-additive capacity, the score of a is computed by means of the following formula: $\sum_{i=1}^{m} m_i a_i + \sum_{i,j,i<j} m_{ij} (a_i \wedge a_j)$ where m_i (resp. m_{ij}) are weights associated with the criteria (resp. the pairs of criteria) and $a_i \wedge a_j$ denotes the minimum of a_i and a_j. There are some constraints on the weights which we do not explicitly state here (see e.g. [3]). The formula for 3-additive capacities is similar; it involves an additional term with weights m_{ijk} associated with triplets of criteria.

To study the questions mentioned in the introduction, the following two experimental settings are brought up. First, to check the expressiveness of the various models, for a given number n of alternatives and m of criteria (or attributes), we randomly generate n alternatives, i.e. n vectors having m components. The values of the m components can be seen as the performances of the corresponding alternative on the m criteria (or attributes). These are randomly drawn from the uniform distribution on the $[0, 1]$ interval. Furthermore, we assume w.l.o.g. that the DM prefers larger values to smaller ones on all attributes. A strict total order on the n alternatives is then drawn randomly from a uniform distribution on all orders. We use linear programming for checking whether this random order on the alternatives can be represented in the various models. In each of the considered models, a score is computed for each alternative, aggregating the performances on the various attributes. Alternative a is judged to be preferred to b if the score of a is greater than the score of b.

Second, to check the behaviour of the Choquet integral models when confronted to non independent preferences, appropriate data and orders are generated according to the same setting as above. To do so, we first randomly generate the performances of two alternatives a and b. Then we draw randomly one attribute index q in $\{1, \ldots, m\}$ and randomly generate two evaluations x_q and y_q. We then construct the four performance vectors (x_q, a_{-q}), (x_q, b_{-q}), (y_q, a_{-q}) and (y_q, b_{-q}), where the notation (x_q, a_{-q}) stands for the profile of alternative a where that evaluation a_q has been replaced by evaluation x_q. We then generate the remaining $n - 4$ alternatives as in the previous setting. The orders are generated such that (x_q, a_{-q}) is ranked before (x_q, b_{-q})

and (y_q, a_{-q}) is ranked after (y_q, b_{-q}), which guarantees that the trials violate the preference independence condition. In practice, we generate a random order on the n alternatives and keep it for the experiment if the previous condition on the 4 profiles is respected; if not, it is rejected.

All the experiments, including the generation of the alternatives and the orders, are performed using R, a free software environment for statistical computing and graphics [9]. The parameters of the first two models are searched for using a linear programming software that can be called from R (lpSolve). The linear program involving the constraints expressing the order on the alternatives that has been randomly generated is submitted to the solver. The structure of the constraints for piecewise linear marginal value functions are inspired by the aggregation-disaggregation approach UTA [6], while the formulation for checking the existence of an additive value function without any restriction on the type of marginal value functions allowed (except that they are non-decreasing) is taken from [5]. For checking the representability by means of a Choquet integral and a 2- or 3-additive capacity, we use the Kappalab R package [3] and, specifically, its "lin.prog.capa.ident" function. In all models, the optimization routine seeks to maximize variable δ, the minimal difference in score of two alternatives that are ranked consecutively in the random order.

We systematically explore the cases of a number of alternatives $n = 4, 5, 6, 8, 10$ and, for each value of n, a number of criteria $m = 3, 4, 5, 6, 8$. For small values of n (4 to 6), we generate between 50 and 100 instances of n alternatives and we systematically try to represent all orders on the alternatives (for the second setting, we consider only the orders which generate non preferentially independent situations). For $n = 8$ and $n = 10$, we generate 50 instances and consider only a random sample of 3000 orders.

3 Results

3.1 Expressiveness of the various models

Table 1 displays the percentage of the orders that can be represented by the various models.

"Add" stands for "additive value function", "Add-3seg" (resp. "Add-2seg") for the particular additive value function model with piecewise linear value functions with 2 segments (resp. 3 segments), "3-cap" for "3-additive capacity", "2-cap" for "2-additive capacity" and "WSum" for "weighted sum". The column "TotNb" shows the total number of trials, "Inst" indicates the number of instances (or performance tables) generated, and "Orders" gives the number of orders generated for each instance. The values between parentheses indicate the percentage of trials which have been prematurely interrupted because they exceeded 3 minutes of calculation time. As an example, for $n = 4$ and $m = 8$, we would expect that the 3-additive capacity model is able to represent a larger number of orders than the 2-additive one. Yet, the lower percentage (80.75) for "3-cap" than for "2-cap" (97.42) is due to the fact that 17.08% of the trials have been prematurely interrupted to avoid too long calculation times. Several timeout values have been tested (from 1 to 15 minutes), however above 3 minutes no significant decrease of the number of interrupted trials could be observed.

The following observations can be made on the data shown in Table 1:

1. For all the models, the proportion of instances that can be represented increases with the number of criteria and decreases with the number of alternatives;

2. The additive value function model is more expressive than the Choquet integral model with a 3-additive capacity; the latter is slightly more expressive than the Choquet integral model with a 2-additive capacity; finally, this method is more expressive than the weighted sum. These differences are more marked when n is large;

3. The difference in expressiveness between the general additive value function model and its derivatives with piecewise linear value functions increases with n. For values of n up to 6 this difference is quite small. For a given value of n, this difference increases with the number of criteria.

One may now wonder how much more expressive a model is when compared to the other ones. Table 2 compares this expressiveness by displaying the percentage of orders that can be represented by a given model while they cannot be by another one. For instance, column "Add/3cap" yields the percentage of orders that can be represented by an additive value function but cannot be represented using a Choquet integral with 3-additive capacities. Let us detail one row of this table. Among all possible orders which were generated for $n = 4$ alternatives and $m = 6$ criteria, 3.88% (resp. 4.38%) can be represented by an additive function but not a Choquet model with a 3-additive (resp. 2-additive) capacity (Add/3cap) (resp. Add/2cap). 13.75% of the orders which are not representable by a weighted sum can be by an additive value function model (Add/Wsum). The 3-additive capacity allows to represent a few more orders than the 2-additive capacity model (0.63%) (3cap/2cap). 9.38% of the orders which cannot be represented by a weighted sum model can be by a 2-additive Choquet integral (2cap/WSum). And none of the orders which cannot be represented by an additive value function can be with a Choquet integral model (3cap/Add and 2cap/Add). Note that the same remark as for

Table 1: Percentage of orders representable in the various tested models.

n	m	TotNb	Inst	Orders	Add (%)	Add-3seg (%)	Add-2seg (%)	3-cap (%)	2-cap (%)	WSum (%)
4	3	2400	100	24	54.54	51.83	48.04	47.00	46.13	35.04
4	4	2400	100	24	73.17	71.04	66.67	68.21	66.29	50.33
4	5	2400	100	24	85.79	83.75	80.92	80.29	78.54	66.38
4	6	2400	100	24	90.58	89.42	87.00	86.71	86.21	76.83
4	8	2400	100	24	98.17	98.17	96.88	80.75 (17.08)	97.42	89.54
5	3	12000	100	120	41.58	36.57	28.97	29.38	27.74	15.00
5	4	12000	100	120	62.58	57.53	50.43	47.46	45.58	28.78
5	5	12000	100	120	78.34	74.09	68.08	68.85	65.78	42.34
5	6	12000	100	120	85.71	81.88	75.73	77.76	75.84	53.33
5	8	12000	100	120	98.00	97.36	94.89	74.92 (21.43)	95.73	79.50
6	3	36000	50	720	25.44	17.66	13.29	11.54	10.66	4.37
6	4	36000	50	720	43.82	34.67	28.18	29.38	26.09	10.82
6	5	36000	50	720	65.63	57.72	48.93	52.88	49.17	22.25
6	6	36000	50	720	80.92	72.54	65.12	68.65	63.39	33.92
6	8	36000	50	720	91.50	89.41	84.32	61.87 (24.70)	84.05	58.04
8	3	150000	50	3000	8.85	3.22	1.60	1.54	1.20	0.23
8	4	150000	50	3000	29.27	16.20	8.34	11.16	7.80	1.05
8	5	150000	50	3000	49.69	31.23	19.70	27.46	20.60	3.38
8	6	150000	50	3000	69.44	50.79	31.25	48.82	34.33	6.71
8	8	150000	50	3000	90.20	80.28	65.20	52.03 (31.71)	71.90	22.04
10	3	150000	50	3000	4.33	0.67	0.18	0.20	0.13	0.01
10	4	150000	50	3000	13.31	3.08	1.01	2.05	1.18	0.04
10	5	150000	50	3000	38.83	12.84	4.37	11.94	5.62	0.21
10	6	150000	50	3000	56.93	24.92	10.69	26.84	15.11	0.72
10	8	150000	50	3000	79.72	55.98	31.19	37.81 (24.70)	42.86	3.86

Table 1 applies here concerning the timeout situations during the search for 3-additive Choquet integral models (figures marked by an asterisk).

The following observations can be made on the data shown in Table 2:

1. There are only very few cases (less than 1%) in which 2- or 3-additive capacity models can represent an order that additive value functions cannot;
2. The advantage of using a 3- instead of a 2-additive capacity is not striking up to $n = 8$ and above.
3. The Choquet integral model can represent significantly more orders than the weighted sum. This difference becomes quite large when the number of criteria is high.

The observations made on basis of Tables 1 and 2 can be confirmed in the following way. Recall that when searching for an additive value model or a capacity, the objective of the linear programs is to maximize the minimal score difference δ between two consecutive alternatives in the ranking. Thus the larger this difference, the easier the model fits the data. Figures 1 and 2 represent histograms of the values of δ obtained by the different models, for $n = 6$ and $m = 8$ (the histograms for other values of n and m are very similar).

In Figure 1 we can see that large values of δ occur more often for the additive value functions than for Choquet integral models. Furthermore, Figure 2 shows that for two segments and three segments value functions there is already a tendency to obtain larger values for δ than with the Choquet integral models (if we consider that the weighted sum is the reference model).

3.2 Choquet integral vs. non preferentially independent data

In Table 2 we observe that there are only very few cases in which 2- or 3-additive capacity models can represent an order that additive value functions cannot. It is well-known (see e.g. [11]) that the Choquet integral allows

Table 2: Comparing expressiveness

n	m	Add/ 3cap	Add/ 2cap	Add/ WSum	3cap/ 2cap	2cap/ WSum	3cap/ Add	2cap/ Add
4	3	7.54	8.42	19.50	0.88	11.08	0.00	0.00
4	4	4.96	6.88	22.83	1.92	15.96	0.00	0.00
4	5	5.50	7.25	19.42	1.75	12.17	0.00	0.00
4	6	3.88	4.38	13.75	0.63	9.38	0.00	0.00
4	8	0.33*	0.75	8.63	0.38*	7.96	0.00*	0.00
5	3	12.20	13.84	26.58	1.64	12.74	0.00	0.00
5	4	15.13	17.01	33.80	1.88	16.79	0.00	0.00
5	5	9.49	12.57	36.00	3.08	23.43	0.00	0.00
5	6	7.95	9.87	32.38	2.02	22.51	0.00	0.00
5	8	1.65*	2.28	18.50	0.51*	16.28	0.00*	0.00
6	3	13.90	14.78	21.08	0.88	6.29	0.00	0.00
6	4	14.56	17.76	33.00	3.29	15.27	0.11	0.03
6	5	12.76	16.47	43.37	3.72	26.91	0.01	0.01
6	6	12.32	17.53	47.00	5.32	29.47	0.05	0.00
6	8	4.94*	7.45	33.46	2.24*	26.05	0.00*	0.00
8	3	7.31	7.65	8.62	0.34	0.97	0.00	0.00
8	4	18.15	21.48	28.22	3.36	6.75	0.04	0.01
8	5	22.26	29.09	46.31	6.86	17.22	0.03	0.01
8	6	20.65	35.12	62.73	14.51	27.62	0.03	0.00
8	8	6.86*	19.93	68.76	8.70*	48.83	0.00*	0.00
10	3	4.13	4.19	4.32	0.06	0.12	0.00	0.00
10	4	11.27	12.13	13.27	0.87	1.14	0.01	0.00
10	5	27.08	33.21	38.62	6.32	5.41	0.19	0.01
10	6	30.17	41.83	56.21	11.73	14.39	0.08	0.01
10	8	16.04*	36.28	76.54	13.39*	40.26	0.00*	0.00

to represent (some) preferences that do not satisfy the preference independence condition, but that satisfy a weaker condition called weak separability. The empirical question that we want to explore is: which proportion of the preferences that do not verify preference independence can be represented by means of a Choquet integral. To test this specifically, we generate data which violate the preference independence condition as mentioned in Section 2 and try to represent these rankings using 2- or 3-additive capacity models. The results are shown in Table 3. For $n = 4, 5, 6$ the number of possible orders is lower than for the first experiment, because of the constraints linked to the generation of non preferentially independent data. Note that the same remark as for Table 1 applies here concerning the timeout situations during the search for 3-additive Choquet integral models

The following observations can be made on basis of the data in Table 3:

1. The percentage of trials representable by a 2- or 3-additive Choquet integral is higher than could be expected from the results in the last two columns of Table 2;
2. The proportion of preference relations that can be represented increases with the number of criteria and decreases with the number of alternatives;
3. Allowing for 3-additive capacities instead of 2-additive capacities results in tiny improvements of the model expressivity.

The first observation above should be both emphasized and commented. Emphasized, because the results in Table 2 seem to indicate that 2-additive and 3-additive capacity models are almost never capable of representing preferences that an additive value function cannot. Actually, these results only show that the way in which the instances are randomly generated seldom yield preferences that are representable by a Choquet integral but not by an additive utility function. In order to specifically sample the set of preferences which cannot be represented by an additive value function, we have introduced two pairs of alternatives specially designed to violate the preference independence condition. In the most favorable experimental conditions we have tested,

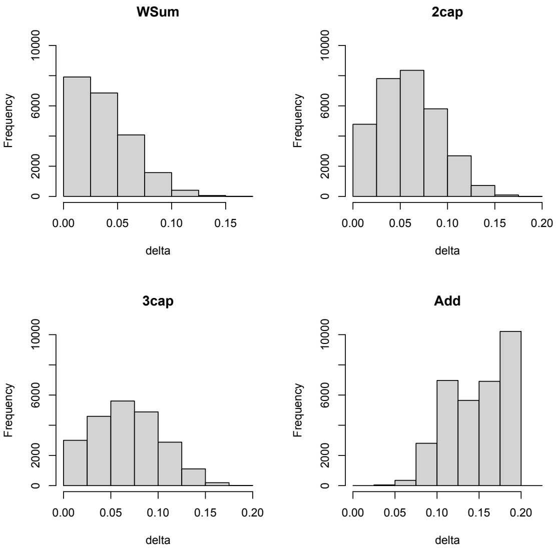


Figure 1: Values of δ: Choquet vs. additive value functions.

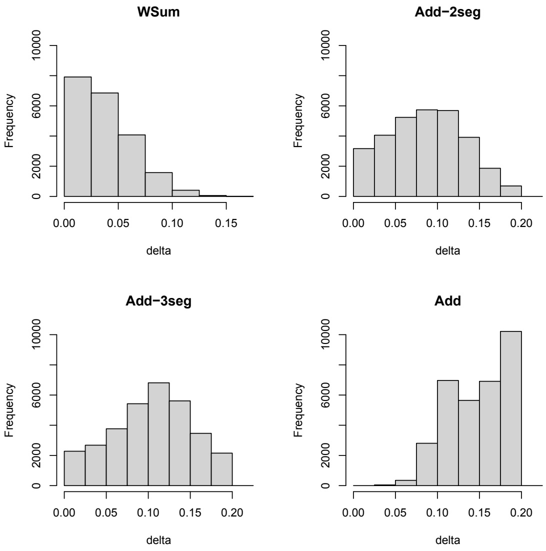


Figure 2: Values of δ: Variants of the additive value functions model.

around 15% of the generated preferences can be represented using 2- or 3- additive capacity models. This figure should however be considered with caution. Indeed the way in which the instances have been generated does not guarantee a uniform sampling of the preferences that cannot be represented in the additive value function model. A precise definition of “uniform sampling” or “fair sampling” is a question that certainly deserves further investigation. We retain as a provisional conclusion that the Choquet integral does have a certain capability of representing preferences that cannot be represented by an additive value function but this capability has still to be precisely assessed.

3.3 Some descriptive information on the data

One might wonder to what extent the violation of the Pareto dominance has an influence on the results presented in Tables 1 and 2. In Table 4 we highlight the dominance situations which occurred during the experiments. First, column “AvgDomPairs” contains the average number of pairs of alternatives which were in a Pareto dominance situation for the generated performance tables. Then, “DomProbsSits” indicates the percentage of trials (performance tables associated with their order) for which a violation of the Pareto dominance has occurred. As a complement, in Figures 3 and 4 we show the relative frequencies of the number of pairs in the Pareto dominance relation for $n = 4$ and $n = 6$ and $m = 3, 4, 6, 8$.

We observe the following points from from Table 4 and Figures 3 and 4:

1. As one would expect, the average number of pairs of alternatives which are in a dominance situation decreases with increasing number of criteria (for n fixed), and increases with the number of alternatives (for m fixed);

2. The percentage of trials violating the Pareto dominance decreases with increasing number of criteria (for n fixed), and increases with the number of alternatives (for m fixed);

3. For high numbers of alternatives and low numbers of criteria, this percentage becomes very large (for example, for $n = 10$ and $m = 3$, we have around 95% of trials which violate the Pareto dominance).

One may also be interested in the trials which are not representable by an additive value function. Their number is given in column “NotAdd” of Table 4 and can easily be obtained by removing from the total number of trials the number of trials which cannot be represented by an additive value function and those which contained at least one violation of the Pareto dominance. As one would expect, the average number of these trials (on the set of possible values for m) seems to increase with n.

4 Conclusion

The additive value function model appears to be definitely more expressive than the Choquet integral (using tractable 2- or 3-additive capacities) at least for the ranges of numbers of alternatives and criteria that we have

Table 3: Non preferentially independent data : percentages of orders representable by the various tested models

n	m	TotNb	Inst	Orders	3-cap (%)	2-cap (%)
4	3	600	100	6	6.83	6.83
4	4	600	100	6	11.00	11.00
4	5	600	100	6	14.33	14.33
4	6	600	100	6	14.83	14.83
4	8	600	100	6	14.67 (1.83)	16.50
5	3	3000	100	30	4.60	4.50
5	4	3000	100	30	9.60	9.57
5	5	3000	100	30	12.40	12.40
5	6	3000	100	30	14.40	14.40
5	8	3000	100	30	12.87 (2.73)	15.53
6	3	9000	50	180	2.90	2.77
6	4	9000	50	180	8.29	7.82
6	5	9000	50	180	11.37	10.99
6	6	9000	50	180	13.03	12.97
6	8	9000	50	180	11.63 (4.10)	15.12
8	3	150000	50	3000	0.79	0.55
8	4	150000	50	3000	2.84	2.17
8	5	150000	50	3000	5.67	4.72
8	6	150000	50	3000	8.27	7.20
8	8	150000	50	3000	9.24 (5.06)	13.43
10	3	150000	50	3000	0.02	0.02
10	4	150000	50	3000	0.63	0.23
10	5	150000	50	3000	2.71	1.54
10	6	150000	50	3000	5.75	3.54
10	8	150000	50	3000	7.07 (4.76)	9.46

explored. These ranges are rather typical in the applications of methods for learning preferences such as UTA [6] or UTA-GMS [5]. It cannot be excluded that our conclusions could be challenged in applications with larger learning sets. However, we have made exploratory trials with 20 alternatives and 8 criteria (using additive value functions, Choquet with 2-additive capacities and weighted sum) resulting in outcomes that are in line with our previous conclusions.

If we now compare the Choquet integral with the weighted sum, we see that using Choquet with 2- or 3-additive capacities may increase the percentage of representable orders (around 15% on average, for 2-additive capacities). The gain of expressiveness w.r.t. the weighted sum seems to be maximal when n is approximately equal to m.

The Choquet integral is sometimes used instead of a sum in the additive value function model [4], obviously increasing the expressiveness of the latter. An indication on how much more expressive such a model is as compared with the simple additive value function model can be obtained by comparing the expressiveness of the Choquet integral with respect to the weighted sum. Indeed, assuming that the distribution of the values of marginal value functions is uniform, we may interpret our randomly generated vectors as representing the marginal values of the alternatives instead of just performances. Under this hypothesis, a weighted sum of these marginal values can be interpreted as an additive value function. Hence, the results of the previous paragraph may be considered as an estimate of what can be gained in terms of expressiveness by considering a Choquet integral instead of a sum in the additive value function model.

In view of these results, one might be tempted to recommend to restrict the use of a Choquet integral to the cases in which marginal value functions have already been elicited and there is strong evidence of interaction between criteria (since the notion of interaction is far from being clear when evaluations have not been recoded into marginal values or at least when the scales of the various criteria have not been made commensurate; see [4], on the notion of commensurateness). We might be less restrictive however if we consider that the most expressive models are not necessarily the best choice in learning models, since they generally involve the specification of many parameters. Indeed, if information about the DM's preferences is scarce, using a less expressive model, such as a 2-additive Choquet integral, may be advocated since it will generally lead to a lower

Table 4: Pareto dominance related information and trials which are not representable by an additive value function model

n	m	TotNb	Inst	Orders	AvgDomPairs	DomProbsSits (%)	NotAdd
4	3	2400	100	24	1.39	45.46	0
4	4	2400	100	24	0.72	26.83	0
4	5	2400	100	24	0.33	14.17	1
4	6	2400	100	24	0.23	9.08	8
4	8	2400	100	24	0.05	1.83	0
5	3	12000	100	120	2.14	58.23	22
5	4	12000	100	120	1.16	37.04	45
5	5	12000	100	120	0.55	21.66	0
5	6	12000	100	120	0.34	14.25	5
5	8	12000	100	120	0.04	2.00	0
6	3	36000	50	720	3.9	74.21	124
6	4	36000	50	720	1.8	54.81	496
6	5	36000	50	720	1.04	33.64	265
6	6	36000	50	720	0.5	19.00	30
6	8	36000	50	720	0.18	8.50	0
8	3	150000	50	3000	7.84	90.95	294
8	4	150000	50	3000	3.74	69.56	1760
8	5	150000	50	3000	1.78	49.79	772
8	6	150000	50	3000	0.82	29.92	949
8	8						
10	3	150000	50	3000	10.6	95.41	402
10	4	150000	50	3000	6.18	85.17	2294
10	5	150000	50	3000	2.48	58.65	3782
10	6	150000	50	3000	1.48	41.87	1813
10	8	150000	50	3000	0.46	19.02	1883

degree of indetermination of the parameters than with an additive utility function. This is all the more true when the DM or the analyst has the intuition that the criteria do interact (although, again, such an "intuition" should be considered with a dose of critical sense). Note that an alternative to general additive value functions is using piecewise linear additive value functions as is done in UTA for instance [6]; such models require the elicitation of fewer parameters than the general additive value function model while their expressiveness is good (Table 1).

References

[1] Bouyssou, D., Pirlot, M.: Preferences for multi-attributed alternatives: Traces, dominance, and numerical representations. J. of Mathematical Psychology 48, 167–185 (2004)

[2] Grabisch, M.: The application of fuzzy integrals in multicriteria decision making. European Journal of Operational Research 89, 445–456 (1996)

[3] Grabisch, M., Kojadinovic, I., Meyer, P.: A review of capacity identification methods for Choquet integral based multi-attribute utility theory; Applications of the Kappalab R package. European Journal of Operational Research 186 (2), 766–785 (2008). DOI 10.1016/j.ejor.2007.02.025

[4] Grabisch, M., Labreuche, C.: Fuzzy measures and integrals in MCDA. In: J. Figueira, S. Greco, M. Ehrgott (eds.) Multiple Criteria Decision Analysis, pp. 563–608. Springer (2004)

[5] Greco, S., Mousseau, V., Slowinski, R.: Ordinal regression revisited: multiple criteria ranking using a set of additive value functions. European Journal of Operational Research 191 (2), 415–435 (2008)

[6] Jacquet-Lagrèze, E., Siskos, Y.: Assessing a set of additive utility functions for multicriteria decision making: the UTA method. European Journal of Operational Research 10, 151–164 (1982)

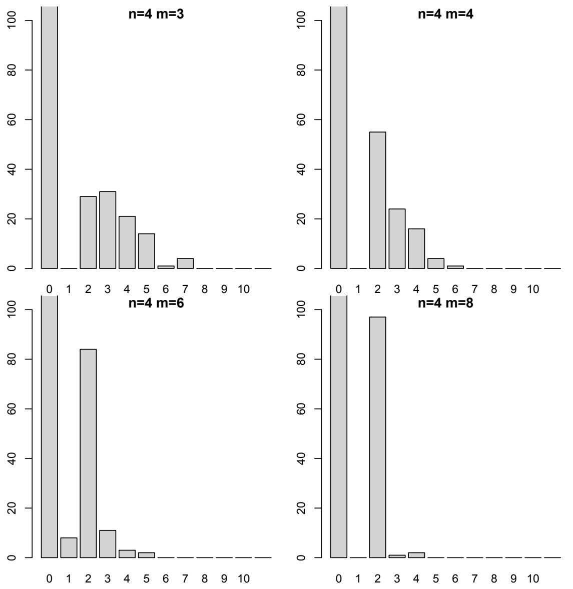


Figure 3: Relative frequencies of the number of Pareto dominance pairs for $n = 4$.

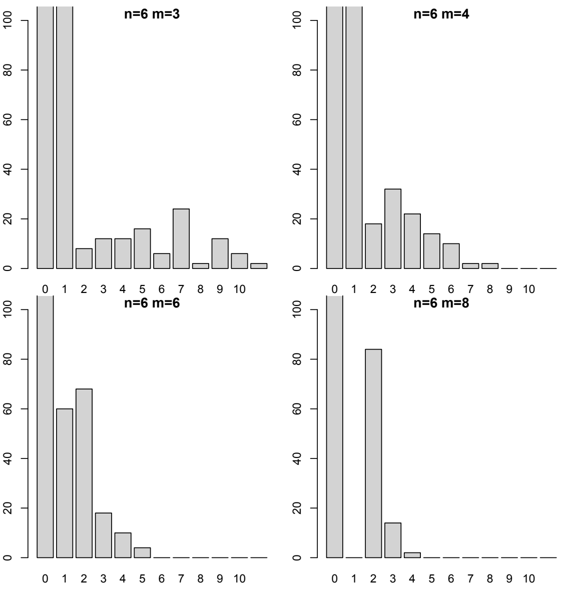


Figure 4: Relative frequencies of the number of Pareto dominance pairs for $n = 6$.

[7] Keeney, R.L., Raiffa, H.: Decision with multiple objectives. Wiley, New-York (1976)

[8] Pirlot, M., Schmitz, H., Meyer, P.: An empirical comparison of the expressiveness of the additive value function and the Choquet integral models for representing rankings. In: 25th Mini-EURO Conference Uncertainty and Robustness in Planning and Decision Making (URPDM 2010). Coimbra (2010)

[9] R Development Core Team: R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria (2005). URL http://www.R-project.org. ISBN 3-900051-00-3

[10] Schmeidler, D.: Integral representation without additivity. Proceedings of the American Mathematical Society 97, 255–261 (1986)

[11] Wakker, P.: Additive Representations of Preferences: A new Foundation of Decision Analysis. Kluwer Academic Publishers, Dordrecht, Boston, London (1989)

Using set functions for multiple classifiers combination

Fabien Rico, Antoine Rolland
Laboratoire ERIC - Université Lumière Lyon 2
5 avenue Pierre Mendes-France
F-69676 BRON Cedex - FRANCE
antoine.rolland@univ-lyon2.fr

Abstract. In machine learning, the multiple classifiers aggregation problems consist in using multiple classifiers to enhance the quality of a single classifier. Simple classifiers as mean or majority rules are already used, but the aggregation methods used in voting theory or multi-criteria decision making should increase the quality of the obtained results. Meanwhile, these methods should lead to better interpretable results for a human decision-maker. We present here the results of a first experiment based on the use of Choquet integral, decisive sets and rough sets based methods on four different datasets.

1 Introduction

A classification problem consists in affecting an individual to a predefined category (or class) from its description via some variables. A classifier or model is a mapping function giving a unique class to each individual. In supervised classification, this function is built from a set of examples thanks to learning method. A large amount of different supervised learning algorithms are used, as indicated for example in [1]. It is then common to have, for a given situation, several results given by several classifiers. These classifiers can be based on the use of very different methods, or can use the same method with variations on leaning set. Using these information to enhance the quality of the classification is the purpose of the multiple classifier aggregation problem. Several methods already exist to solve the multiple classifier aggregation problem (see [11] and [12] for a survey). We propose here new aggregation procedures inspired by some aggregation methods developed in the framework of multi-criteria decision making and social choice theory. In section 2, we present the multiple classifier aggregation problem and stand the needed notations. In section 3 we briefly present some aggregation procedures based on the use of set functions and their applications in our framework; then in section 4 we present the implementations and tests of these methods and propose an analysis of the results.

2 Multiple classifier aggregation

2.1 Multiple classifier

It is well known that there exists no perfect classifier neither universal classifier : each classifier makes mistakes, and each classification algorithm is really performing only on specific situations. So in order to reduce the errors number, it should be interesting to mix the results of several classifiers. Given a specific classifier, we can increase its performances by adding one or several other classifiers. These new classifiers should be as independent as possible from the first one to be able to ‘correct” its errors. It is the case for example in the boosting method where classifiers are built to obtain a maximum diversity [15]. But the new classifiers should also be intrinsicaly performant, although they will degrade the general performance. On the other hand, if the first classifier is still good, many other good classifiers will be strongly related to the first one : it should be difficult to find another good classifier independent from the first one. Therefore, adding a new classifier will not give much more information to the decision maker. So two mains properties have to be considered for selecting and aggregating classifiers: the quality of each classifier and the diversity into the set of classifiers. Mean rules and majority rules are very dependent of the quality of the added classifier for one hand, and of the independence between classifiers on the other hand (see [14] for a theoretical study). We investigate in this paper some other aggregation procedures which should manage less quality and/or dependent classifiers.

2.2 Aggregation procedures

There already exist several aggregation procedures for the multi-classifier problem [11]. Two of them are considered as reference procedures, due to their use facility, and the fact that they are easily understandable :

- the majority rule : the allocated class for an individual is the class chosen by a majority of classifiers.
- the mean rule : the allocated class for an individual is obtained by a cutting level applied on the mean of the different labels given by each classifiers.

In this paper, we present new aggregation procedures which aim at enhancing the quality of these two procedures. The general idea is that a multiple classifiers aggregation procedure can be seen as a particular case of either a voting procedure, or a multi-criteria aggregation rule, as seen below :

- suppose that each classifier is a voter, who can vote *for* or *against* allocating x in class a. Then the aggregation of classifiers problem can be seen as a voting procedure.
- suppose that each classifier is giving a score related to the strength of its conviction that individual x should be affected to class 1. This score can be seen as a value taken by a criterion related to the considered classifier. Then the multiple classifiers aggregation can be seen as a multi-criteria aggregation problem.

The field of multi-criteria aggregation procedure or voting procedure has been well studied in the past decades, and several approaches and methods are available solve these aggregation problems in social choice theory or multi-criteria decision aiding theory (see [2] for a review). We will focus here on a few a them, based on the same basic tool which is the use of set functions to represent the importance of each coalition of voters (resp. criteria).

2.3 Notations

We first establish the needed notations to have a formal representation of our framework. We define formally a classifier aggregation problem as a problem which consists in aggregating the information given by m classifiers on a individual ω in order to sort him into a pre-defined class. We note here Ω the set of n individuals $\{\omega_1, \ldots, \omega_n\}$ to be classified. Each individual ω_i is described by a set of q predictor variables $X^j \in \mathcal{R}$, $j = 1, \ldots, q$ and a class of membership $Y \in \{0, 1\}$. By convention for the i^{th} individual ω_i we denote Y_i its class and $X_i = (X_i^1, \ldots, X_i^q) \in \mathcal{R}^q$ its representation.

A single classifier ϕ is a mapping :

$$\begin{array}{rccc} \phi: & \mathcal{R}^q & \rightarrow & [0,1] \\ & X & \mapsto & \phi(X) = p \end{array}$$

The result $\phi(X)$ is said to be a label according to X by the classifier ϕ. It can be seen as a score (probability, possibility...) for individual ω represented by X to belong to class 1.

According to this classifier, the chosen class should be obtained by a cutting level α :

$$c(X) = 1 \text{ if } \phi(X) > \alpha$$

Let $\mathcal{P} = \{\phi_1, \ldots, \phi_m\}$ be a set of m classifiers. An individual X can then be described by a vector of labels given by each classifiers $p_x = (p_1, \ldots, p_j, \ldots, p_m)$, or by a vector of chosen class affected by each classifier $c_x = (c_1, \ldots, c_j, \ldots, c_m)$.

The multiple classifier aggregation problem consists in aggregating the m outputs of the m classifiers to get a unique chosen class C.

An multi-classifier aggregation function Φ is a mapping :

$$\begin{array}{rccc} \Phi: & [0,1]^m & \rightarrow & \{0,1\} \\ & \{\phi_i(X)\} & \mapsto & \Phi(X) = C \end{array}$$

As $c_j(X) \in \{0, 1\}$ $\forall \phi \in \mathcal{P}$ also, the multi-classifier aggregation function Φ can also takes a vector c_x as argument.

2.4 General framework

We focus here on aggregation problems with a few number of different classifiers (typically less than 10 classifiers). The input of the aggregation procedure is a vector $p_x = (p_1, \ldots, p_m) \in [0, 1]^m$ of labels or a vector $c_x = (c_1, \ldots, c_m) \in \{0, 1\}^m$ of classes. The result is a unique chosen class C_X.

A classifier gives for each individual a class which can be wrong or right, as soon as the real class Y of the individual is known. Let us recall that four situations can happen with a classifier. The following table stands the different sets cardinals for each possibility :

	real class	
obtained class	a	b
a	n_{aa}	n_{ab}
b	n_{ba}	n_{bb}

The quality of a classifier can be measured by several indicts.

- success ratio, denoted su.

$$su = \frac{n_{aa} + n_{bb}}{n}$$

The success ratio is the ratio of the number of well-affected individuals divided by the total number of individuals. It measures the ability of the classifier to well classify the individuals, whatever their class should be.

- precision ratio for the class a, denoted pr_a.

$$pr_a = \frac{n_{aa}}{n_{aa} + n_{ab}}$$

The precision ratio is the number of well-affected individuals of class a on the total number of individuals affected by the procedure to the class a. It measures the ability of the classifier to well reject the individuals which are not supposed to belong to the class a.

- callback ratio for the class a, denoted cr_a :

$$cr_a = \frac{n_{aa}}{(n_{aa} + n_{ba})}$$

The callback ratio is the ratio of the number of well-affected individuals of class a divided the total number of individuals of class a. It measures the ability of the classifier to well detect the individuals of class a : it is an asymmetric ratio, which is rather used in the field of statistic tests, or disease detection.

3 Set functions approaches

As mentioned in section 2.2, the multi-classifier aggregation problem has strong formal links with the preference aggregation problem in social choice theory or multi-criteria decision making. Considering each classifier as a voter, we wonder if there exist some coalitions (sets of classifiers) such that if all the classifiers of a coalition agree on class a for individual ω then the aggregation result of $\Phi(X)$ is class a. We would like to represent the existence of such coalitions through set functions, roughly giving to each subset of $\mathcal{P}$ a weight corresponding to its power as a coalition. We present in this paper three methods based on a decisive sets concept.

3.1 Capacity and Choquet Integral

3.1.1 Definition

One of the limits of the use of the weighted mean as an aggregation function is that it is unable to take into account synergy possibly happening between criteria to aggregate. A Choquet integral (see [5], [13] for a complete presentation) can then be seen as a non-additive generalization of the weighted mean. It is based on the use of a non-additive set function named capacity :

Definition 1. *Let N be a set of objects and $\mu = card(N)$. A capacity $v : 2^N \rightarrow \mathbb{R}^+$ is a set function such that $v(\emptyset) = 0$, and $A \subseteq B \subseteq N$ implies that $v(A) \leq v(B)$. A capacity is said to be normalized iff $v(N) = 1$.*

Formally, a Choquet integral is a function $\mathcal{C}$ from $[0, 1]^\mu$ into $[0, 1]$ such that, $\forall x = (x_1, \ldots, x_\mu) \in [0, 1]^\mu$:

$$\mathcal{C}(x) = \sum_{i=1}^{\mu} x_{\sigma(i)}(v(A_{\sigma(i)}) - v(A_{\sigma(i-1)}))$$

where

- σ is a permutation on $\{1,\dots,\mu\}$ such that $\sigma(1) \leq \sigma(2) \leq \dots \leq \sigma(\mu)$
- v is a capacity on the set $\{1,\dots,\mu\}$.
- $A_{\sigma(i)} = \{\sigma(i), \sigma(i+1), \dots, \sigma(\mu)\}$

The Choquet integral has been very used in the fields of decision under uncertainty and multi-criteria decision aiding along the past decade, as mentioned in [7].

	Choquet integral based aggregation rule
Input	a set of individuals *App*
	$p_X = (\phi_1(X),\dots,\phi_m(X)\ \forall\omega \in App$
	or $c_X = (c_1(X),\dots,c_m(X))\ \forall\omega \in App$
	$Y_\omega\ \forall\omega \in App$
Output	a capacity v on the set $\{1,\dots,m\}$
Aggregation	$C(X) = 1 \iff \alpha < \Phi(X)$
	$\Phi(X) = \sum_{i=1}^{m} \phi_{\sigma(i)}(X)(v(A_{\sigma(i)}) - v(A_{\sigma(i-1)}))$

Table 1. Summary of Choquet integral model

3.1.2 *Analogy with the multi-classifier aggregation problem*

Each classification function ϕ_i is giving a label in $[0,1]$ to the individual ω. Formally, each classifier can then be seen as a criterion and the function Φ as an aggregation function on these criteria. If a capacity function is defined on the set of classifiers, we can then use a Choquet integral as an aggregation function to obtain a global score for individual ω described by predictor variables X. We obtain, with the above notations,

$$\Phi'(X) = \sum_{i=1}^{m} \phi_{\sigma(i)}(X)(v(A_{\sigma(i)}) - v(A_{\sigma(i-1)}))$$

where

- σ is a permutation on $\{1,\dots,m\}$ such that $\sigma(1) \leq \sigma(2) \leq \dots \leq \sigma(m)$
- v is a capacity on the set $\{1,\dots,m\}$.
- $A_{\sigma(i)} = \{\sigma(i), \sigma(i+1), \dots, \sigma(m)\}$

The chosen class should then be obtained from Φ' by a cutting level α.

3.1.3 *Using Choquet integral in multi-classifier aggregation framework*

The aim of the use of a Choquet integral in a multi-classifier aggregation problem is to exhibit interactions which can appear between classifiers. In order to do so, we will use identification procedures based on a least square approach as proposed in [6]. These procedures use a learning set of individuals as input. The label vector p_X for each individual given by all the classifiers is known, such as the real class of each individual, and the identification procedure is an optimization program that compute the parameters of the Choquet integral that better fit the learning set. We then use the calculated parameters to infer the category of new individuals.

We implemented two procedures:

- *Choquet ls* uses least-square based approach to infer the parameters of the whole set of capacity values.
- *Choquet 3-add* uses least-square approach also but is limited to a 3-additive capacity, i.e. a capacity with no interactions between sets of more than 3 criteria (see [4] for details on *k*-additivity). This limit has been chosen as a compromise, in order to facilitate the computation as it divides by two the number of parameters, but keeping a relevant amount of interaction between criteria.

The first experiments show that between 50 and 85% of the Mbius coefficients are almost null. For example, we can have $v(\{1\}) = 0$, $v(\{2\}) = 0$ and $v(\{1,2\}) = 1$. It means in that case that if a alternative is classified in class 1 for both classifiers 1 and 2, then it should be classified in class 1 by the Choquet Integral operator. Note that it is not always easy to obtain such a simple semantic interpretation of the capacity parameters.

It is not always easy to obtain such a simple semantic interpretation of the capacity parameters and we have not study thoroughly the results. However, the first experiments show that between 50 and 85% of the Mbius coefficients are almost null. For example, we can have as typical parameters $v(\{1\}) = 0$, $v(\{2\}) = 0$ and $v(\{1,2\}) = 1$. It means in that case that if a alternative is classified in class 1 for both classifiers 1 and 2, then it is classified in class 1 by the Choquet Integral operator. This may be compared to the decisive set method described below, noting that the Choquet integral method can take into account both positive and negative examples in learning.

3.2 Decisive sets

3.2.1 *Definition*

In social choice theory, voters $v_1,\dots,v_n$ are supposed to be able to give a preference relation between two candidates (or individuals) x and y. The fact that voter v_1 prefers candidate x to candidate y is denoted by $x \succ_{v_1} y$. Following Fishburn [3], a voter v_i is said to be *decisive for the pair* (x,y) if the fact that $x \succ_{v_i} y$ implies that x is preferred to y in the aggregated order, denoted $x \succ y$. A voter who is decisive for all pair x, y is said to be totally decisive, or just *decisive*. Inspired by Weymark [17], we can also define a decisive set of voters $V = \{v_i,\dots,v_j\}$ for the pair (x,y) if the fact that $x \succ_{v_i} y$ $\forall v_i \in V$ implies that $x \succ y$.

	Decisive sets based aggregation rule
Input	a set of individuals *App*
	$c_X = (c_1(X),\dots,c_m(X))\ \forall\omega \in App$
	$Y_\omega\ \forall\omega \in App$
Output	$\mathcal{D}$, a set of K decisive subsets
	$D_k \subseteq \mathcal{P},\ k = 1,\dots,K$ for the class a
Aggregation	$C(X) = a \iff \exists D \in \mathcal{D}$
	such that $\{i \in 1,\dots,M \mid C_i(X) = a\} \subseteq D_k$

Table 2. Summary of Decisive sets model

3.2.2 *Analogy with the multi-classifier aggregation problem*

Analogously, we can settle the following definitions in our framework:

Definition 2. *A classifier $\phi_i \in \mathcal{P}$ is said to be* decisive for *X for the class a if $c_i(X) = a \Rightarrow C(X) = a$. If ϕ_i is decisive for all X, ϕ_i is said to be totally decisive, or simply* decisive.

Definition 3. *a set of classifiers $P \subseteq \mathcal{P}$ is said to be* decisive for *X for the class a if $\forall\phi_i \in P$, $c_i(X) = a \Rightarrow C(X) = a$. If P is decisive for all X, P is said to be totally decisive, or simply* decisive.

3.2.3 *Using decisive sets in multi-classifier aggregation framework*

Practically, the aim of the identification process is to discover a set of decisive sets as small as possible for a given class a. In order to identify these decisive sets, we study a learning set of known individuals and we first catch all the existing decisive sets for each individual. Then we select the smallest (for the inclusion) decisive sets of classifiers that optimize the chosen ratio. We then use this set of decisive sets to infer the category of new individuals. The choice of $a = 0$ or $a = 1$ and the choice of the good ratio as an indicator of the fit quality have an importance on the detected decisive sets. We present below results obtained by considering successively $a = 0$ (method *Decisive sets 0*) or $a = 1$ (method *Decisive sets 1*) both focusing on the success ratio.

3.3 Rough sets dominance-based approximation

3.3.1 *Definition*

Another approach consists in using rough sets through the dominance-based rough set approach (see Greco, Matarazzo and Slowinski [9], [10]). In multi-criteria decision aiding, this approach uses decision rules to assign the alternatives to the different categories, with respect to some reference levels on each criterion. The axiomatic foundations of the rough set approach have been well studied by Greco, Mattarazo and Slowinski, including characterization of the sorting problem using a utility function or an outranking relation [8] or a Sugeno integral [16]. The dominance-based rough set approach for classification consists first in obtaining for each alternative the set of all the classes compatible with the dominance relation on the alternatives. It then produces a set of decision rules which characterize the allocation of each alternative to the possible classes. Decision rules present themselves as "if the value of the alternative on criteria i is at least . . . and the value of the alternative on criteria j is at least . . ., then the category of the alternative is at least . . .."

dominance-based rough sets based aggregation rule	
Input	a set of individuals App
	$c_X = (c_1(X), \dots, c_m(X))\ \forall \omega \in App$
	$Y_\omega\ \forall \omega \in App$
Output	$\mathcal{D}$, a set of K decisive subsets
	$D_k \subseteq \mathcal{P},\ k = 1, \dots, K$ for the class a
Aggregation	$C(X) = a \iff \exists D \mathcal{D}$
	such that $\{i \in 1, \dots, M \mid C_i(X) = a\} \subseteq D_k$

Table 3. Summary of dominance-based rough sets model

3.3.2 *Analogy with the multi-classifier aggregation problem*

Each classification function ϕ_i is giving a score on $[0, 1]$ for the individual ω. Formally, each classifier can then be seen as a criterion and each individual as an alternative. Each alternative can then be classified only in one out of two classes. A dominance-based rough sets approach will then consist in sorting each individual into one out of three classes : individuals which are certainly in class a, individuals which are certainly not in class a, and ambiguous individuals, based on the dominance relation between individuals on values $\phi_i(X)$. We have then to produce a decision rules set to characterize the allocation of each individual to class 0 or 1. We can also directly use the classification vector c_X in the dominance-based rough sets approach. All the variables are then binary variables, and then decision rules can be interpreted as decisive sets of classifiers. We will then focus on this case.

3.3.3 *Using dominance-based rough set approach in multi-classifier aggregation framework*

Following the analogy developed in the decisive sets frameworks, we decide to aggregate the results c_X of the classifiers to obtain the final class for individual X. The inputs of the procedure are then only binary vectors $c_X = (c_1(X), \dots, c_m(X))$ with $c_i(X) \in \{0, 1\}$. The use of a dominance-based rough set approach in multi-classifier aggregation consists simply in finding a set of decision rules that better fits the learning set of individuals. Decision rules present themselves as "if $c_i(X) = a$ and . . . and $c_j(X) = a$ then $c(X) = a$". These rules can also be interpreted as decisive sets of classifiers : "if $c_i(X) = a$ and . . . and $c_j(X) = a$ then $c(X) = a$" means that $\{\phi_i, \dots, \phi_j\}$ is a decisive set for class a. The used algorithm consists in building decisive sets from an empty set of classifiers, adding new classifiers in the set while the chosen ratio keeps on being optimized. The choice of $a = 0$ or $a = 1$ and the choice of the good ratio as an indicator of the fit quality have an importance on the detected decision rules. We present below results obtained by considering successively $a = 0$ (method *Rough sets 0*) or $a = 1$ (method *Rough sets 1*) both focusing on the success ratio.

4 Results

4.1 Data sets

We have compared those aggregation methods versus majority and mean rules for the following four datasets:

- UCI's dataset `Letter`: recognition of letter "R" versus "B".
- UCI's dataset `Musk (v2)`: prediction if a molecule is (or not) a musk.
- Leo Breiman's `Ringnorm` and `Threenorm`: recognition of two normal distribution with different mean and covariance.

Those datasets have medium size (detailed in table Tab:datasets) from 1500 to 6600 individuals), which gives sufficient individuals for the two learning steps (training simple classifiers and training aggregating methods). They have 2 classes and two of them are real examples (Letter and Musk) while the others (Threenorm and Ringnorm) are constructed data.

	nb indiv.	nb var.	prop of 1
Letter	1524	16	49.7%
Musk	6599	166	84%
Ringnorm	2128	20	50%
Threenorm	2128	20	50%

Table 4. List of the considered datasets.

4.2 Compared methods

We have compared the error, precision and call-back ratios through the three different methods for the four datasets. In order to do so, we split each dataset into a learning set L and a test set T. The learning set has been used to train the classifier and the test one to compare the computed class with the true one. More, our method used two levels of training, one for the simple classifiers to build m models and one for the aggregation model. So for our algorithm, the learning set L is itself split in two equal parts L_{train} and L_{agg}.

- L_{train} is used for training 7 simple well-known classifiers:
  - Breiman's random forest from the `randomForest` R library;
  - ada boost from the `ada` R library;
  - support vector machine using C classification and Gaussian kernel (`ksvm` function from `kernlab` R package);
  - linear Discriminant Analysis from `MASS` R package;
  - logistic regression using `glm` from `stats` R package;
  - single decision tree C4.5 using `J48` function provided by `RWeka` R package;
  - k nearest neighbours using `IBk` function from `RWeka` R package by default (k=1).

  Then we obtain $$\begin{array}{rcl} p: \quad \Omega & \to & [0,1]^m \\ x & \mapsto & p_x = (p_1, \ldots, p_m) \end{array}$$
- The responses of the obtained classifiers are computed on L_{agg} and T, to obtain respectively the $p(L_{agg})$ and $p(T)$ results.
- The aggregation operator is trained using classifiers responses $p(L_{agg})$ and true classes $Y(L_{agg})$ in order to obtain the multi-classifier aggregation function Φ.
- The aggregated response for test set $\Phi(p(T))$ is computed and compared to the true class $Y(T)$ to compute the different ratios.

For mean and majority aggregation, the two levels learning is not necessary, so the classifier's training process is done one more time using the entire learning set L.

We also use Wilcoxon signed rank test to detect if the differences are significant or not. Our several learning sets and test sets are computed using 10 cross-validations. This means that the dataset is divided into 10 disjoint parts. We repeat the same test 10 times, each time, one part is used as test set and one is used for learning algorithm. The presented ratios are the means of the 10 corresponding results and the significance of the differences is computed thanks to Wilcoxon test.

4.3 Results

We present in tables 5 to 8 the results of our experiment on the different datasets. For each dataset, we present success ratio for each aggregation method, precision and callback ratio for class 1. For each aggregation method we indicate the significance degree (with $\alpha = 5\%$) compared first to the mean rule and second to the majority one.

- "+" denotes that the proposed aggregation method is significantly better than mean (respect. majority) rule,
- "-" denotes that it is significantly worse than mean (respect. majority) rule,
- "=" denotes that the difference is not significant.

For example in table 8, the success ratio of rough set oriented for 0 class is 87.1%, which is significantly better than majority rule but not than mean rule.

We can see that aggregation methods are often better than majority or mean rule, rarely worst (and never for success ratio). These results are promising as they are obtained with non optimized algorithm. For example we haven't study the effect of the size of L_{train} and L_{agg}, choosing same size for the both. This means that simpler classifiers (majority and mean rule) are trained on 2 times bigger sets. Our first intuition was that the orientation of decision or rough sets research should have an effect on precision and callback ration, but this is not obvious in our experiments. However, further studies in this direction certainly need to be lead.

Agg. method	Success ratio	Precision ratio	Call-back ratio
Mean	98.7	98.8	98.5
Majority	98.8	99.1	98.5
Decisive sets 1	98.6 =/=	99.0 =/=	98.2 =/=
Decisive sets 0	98.7 =/=	98.8 =/=	98.5 =/=
Rough sets 1	98.6 =/=	99.5 =/=	97.7 =/=
Rough sets 0	98.5 =/=	98.4 =/=	98.5 =/=
Choquet ls	98.8 =/=	99.2 =/=	98.4 =/=
Choquet 3-add	98.8 =/=	99.1 =/=	98.5 =/=

Table 5. Comparison of several methods for the Letter R/B data set

Agg. method	Success ratio	Precision ratio	Call-back ratio
Mean	97.6	97.7	99.5
Majority	97.7	97.8	99.6
Decisive sets 1	97.9 =/=	98.0 =/=	99.6 =/=
Decisive sets 0	98.1 +/+	98.6 +/+	99.2 -/-
Rough sets 1	98.2 +/+	98.5 +/+	99.3 -/=
Rough sets 0	98.2 +/+	98.4 +/+	99.5 =/=
Choquet ls	98.2 +/+	98.7 +/+	99.2 -/-
Choquet 3-add	98.2 +/+	98.7 +/+	99.1 +/+

Table 6. Comparison of several methods for the Musk data set

Agg. method	Success ratio	Precision ratio	Call-back ratio
Mean	95.9	94	98.3
Majority	94.2	92.5	96.2
Decisive sets 1	98.4 +/+	98.1 +/+	98.7 =/+
Decisive sets 0	97.2 =/=	98.3 +/+	96.0 =/=
Rough sets 1	98.5 +/+	98.2 +/+	99.0 =/+
Rough sets 0	92.8 =/=	90.1 =/=	99.4 +/+
Choquet ls	98.4 +/+	98.2 +/+	98.7 =/+
Choquet 3-add	98.4 +/+	98.2 +/+	98.7 =/+

Table 7. Comparison of several methods for the Ringnorm data set

5 Conclusion

In this paper, we obtained promising results which need further investigations. Among others, we propose two issues which are in our opinion relevant to be study:

- Does this approach can be applied to a larger number of classifiers ? This will be interesting to use it in ensemble methods framework, where several tens (or hundreds) of classifiers are aggregated. This leads to computation problems, because the complexity of some methods grows exponentially with the number of simple classifiers.

Agg. method	Success ratio	Precision ratio	Call-back ratio
Mean	85.8	86.1	85.4
Majority	86.	85.9	86.3
Decisive sets 1	86. =/=	83.7 =/-	89.9 +/+
Decisive sets 0	86.5 =/=	87.1 =/+	85.7 =/=
Rough sets 1	86.6 =/=	88.1 +/+	84.7 =/-
Rough sets 0	87.1 =/+	85.6 =/=	89.3 +/+
Choquet ls	87.5 +/+	87.4 =/+	87.7 =/=
Choquet 3-add	87.6 +/+	87.5 +/=	87.8 =/=

Table 8. Comparison of several methods for the Threenorm data set

- May these methods be used for selecting classifiers ? Indeed, rough set methods give generally a small number of rules. This may be seen as a simplification of the original set of classifiers. One drawback of aggregating different classifiers is that the process disintegrate the decision in multiple classifier, making it impossible to understand. So a human decision maker may need such a simplification.

REFERENCES

[1] Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Information Science and Statistics, Springer, 2006.

[2] *Concepts and Methods of Decision-Making*, eds., Denis Bouyssou, Didier Dubois, Marc Pirlot, and Henri Prade, Wiley-ISTE, 2009.

[3] P. C. Fishburn, *The Theory of Social Choice*, Princeton University Press, 1973.

[4] M. Grabisch, 'k-order additive discrete fuzzy measures and their representation', *Fuzzy Sets and Systems*, **92**, 167?189, (1997).

[5] M. Grabisch and M. Roubens, 'Application of the Choquet integral in multicriteria decision making', in *Fuzzy Measures and Integrals-Theory and Applications*, eds., M. Grabisch, T. Murofushi, and M. Sugeno, 348–374, Physica Verlag, (2000).

[6] Michel Grabisch, Ivan Kojadinovic, and Patrick Meyer, 'A review of methods for capacity identification in choquet integral based multi-attribute utility theory: Applications of the kappalab r package', *European Journal of Operational Research*, **186**(2), 766–785, (2008).

[7] Michel Grabisch and Christophe Labreuche, 'A decade of application of the choquet and sugeno integrals in multi-criteria decision aid', *4OR: A Quarterly Journal of Operations Research*, **6**, 1–44, (2008).

[8] S. Greco, B. Matarazzo, and R. Slowinski, 'Conjoint measurement and rough set approach for multicriteria sorting problems in presence of ordinal criteria', in *A-MCD-A, Aide Multicritère à la Décision/Multiple Criteria Decision Aid*, eds., A. Colorni, M. Paruccini, and B. Roy, 117–144, European Commission, Joint Research Centre, EUR 19808 EN, Ispra, (2001).

[9] S. Greco, B. Matarazzo, and R. Slowinski, 'Rough sets theory for multicriteria decision analysis', *European Journal of Operational Research*, **129**, 1–47, (2001).

[10] S. Greco, B. Matarazzo, and R. Slowinski, 'Rough sets methodology for sorting problems in presence of multiple attributes and criteria', *European Journal of Operational Research*, **138**, 247–259, (2002).

[11] L. I Kuncheva, *Combining Pattern Classifiers. Methods and Algorithms*, Wiley, 2004.

[12] Ludmila I. Kuncheva. Classifier ensembles: Facts, fiction, faults and future, 2008. (slides, plenary talk).

[13] J.-L. Marichal, 'An axiomatic approach of the discrete Choquet integral as a tool to aggregate interacting criteria', *IEEE Transactions on Fuzzy Systems*, **8**(6), 800–807, (December 2000).

[14] Dymitr Ruta and Bogdan Gabrys. A theoretical analysis of the limits of majority voting errors for multiple classifier systems, 2000.

[15] Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods, 1997.

[16] R. Slowinski, S. Greco, and B Matarazzo, 'Axiomatization of utility, outranking and decision-rule preference models for multiple-criteria classification problems under partial inconsistency with the dominance principle', *Control and Cybernetics*, **4**(31), 1005–1035, (2002).

[17] J. A. Weymark, 'Arrow's theorem with social quasi-orderings', *Public Choice*, (42), 235–246, (1984).

Eyke Hüllermeier, Department of Mathematics and Computer Science, Philipps-Universität Marburg, Germany

"***Preference Learning : an Introduction***",

The topic of "preferences" has recently attracted considerable attention in artificial intelligence in general and machine learning in particular, where the topic of preference learning has emerged as a new, interdisciplinary research field with close connections to related areas such as operations research, social choice and decision theory. Roughly speaking, preference learning is about methods for learning preference models from explicit or implicit preference information, typically used for predicting the preferences of an individual or a group of individuals. Approaches relevant to this area range from learning special types of preference models, such as lexicographic orders, over "learning to rank" for information retrieval to collaborative filtering techniques for recommender systems. The primary goal of this tutorial is to survey the field of preference learning in its current stage of development. The presentation will focus on a systematic overview of different types of preference learning problems, methods and algorithms to tackle these problems, and metrics for evaluating the performance of preference models induced from data.

Eyke Hüllermeier, Department of Mathematics and Computer Science, Philipps-Universität Marburg, Germany

"***Preference Learning : an Introduction***",

The topic of "preferences" has recently attracted considerable attention in artificial intelligence in general and machine learning in particular, where the topic of preference learning has emerged as a new, interdisciplinary research field with close connections to related areas such as operations research, social choice and decision theory. Roughly speaking, preference learning is about methods for learning preference models from explicit or implicit preference information, typically used for predicting the preferences of an individual or a group of individuals. Approaches relevant to this area range from learning special types of preference models, such as lexicographic orders, over "learning to rank" for information retrieval to collaborative filtering techniques for recommender systems. The primary goal of this tutorial is to survey the field of preference learning in its current stage of development. The presentation will focus on a systematic overview of different types of preference learning problems, methods and algorithms to tackle these problems, and metrics for evaluating the performance of preference models induced from data.

Poster session

- "*Preference Learning to Rank : An Experimental Case Study*", M. Abbas, USTHB, Alger, Algerie

- "*From preferences elicitation to values, opinions and verisimilitudes elicitation*", I. Crevits, M. Labour, Université de Valenciennes,

- "*Group Decision Making for selection of an Information System in a Business Context*", T. Pereira, D.B.M.M Fontes, Porto, Portugal

- "*Ontology-based management of uncertain preferences in user profiles*", J. Borras, A. Valls, A. Moreno, D. Isern, Universitat Rovira i Virgili, Tarragona

- "*Optimizing on the efficient set. New results*", D. Chaabane, USTHB, Alger, Algerie

From preferences elicitation to values, opinions and verisimilitudes elicitation

Igor Crévits (a), Michel Labour (b)
Université de Valenciennes et du Hainaut-Cambrésis
(a) Laboratoire d'Automatique de Mécanique et d'Informatique industrielles et Humaines
(b) Design VISuel et Urbain
59313 valenciennes Cedex 9
Igor.Crevits@univ-valenciennes.fr, mlabour@gmail.com

Introduction

Methodological reflections are rare in the decision aiding domain. The most striking works on the subject show the basic features of the decision analyst's craft. The development from *Multicriteria Methodology for Decision Aiding* (MCMDA) [Roy1985] to *Decision Aiding Process* (DAP) [Tsoukiàs2007] is done by including a representation of a designated problem. This representation initiates a request for a decision aid, ascribed as the Problem situation. Such decision aiding is facilitated by a mathematical model, called Evaluation model. Several possible decisional choices are gathered in the Problem formulation. This is numerically evaluated by the Evaluation model in order to reduce the amount of possible choices and to generate a Final recommendation. However, the passage from the identification of a Problem situation to establishing a Problem formulation is not evaluated.

Given this, we argue for the insertion of a Problem-based evaluation model (denoted by $\mathcal{M}_{\mathcal{P}}$) in DAP. The constituting elements of $\mathcal{M}_{\mathcal{P}}$ are based on the definition of a decision problem, seen as an aggregation of values, opinions and likelihoods [Colorni2012]. This leads us to advance a matrix-like representation for $\mathcal{M}_{\mathcal{P}}$, called Decision grid. The construction of the Grid is linked to preferences elicitation that conjointly extends to a definition of potential actions and criteria development. The Grid produces building blocks useful in the construction of the Evaluation model. This done, we then illustrate our approach in the domain of air traffic control.

Methodological frameworks of decision aiding

The founding father of Multicriteria Decision Aiding Methodology, Bernard Roy writes:

"Decision aiding is the activity of the person who, through the use of explicit but not necessarily completely formalized models, helps obtain elements of responses to the questions posed by a stakeholder of a decision process. These elements work towards clarifying the decision and usually towards recommending, or simply favouring, a behaviour that will increase the consistency between the evolution of the process and the stakeholder's objectives and value system" [Roy1985].

Roy's definition also applies for the definition of the idea of "decision". In using a constructive approach, decision aiding can this be seen as a decision linked to a mathematical model. To build this model, [Roy1985] advances MCMAD led by the analyst that develops a decision aiding framework.

The grassroots participants, concerned by the decision process, are involved in the framework development at levels I and II in the construction of potential actions and criteria. These two levels converge towards a numerical representation of a common decision for the participant and the analyst. Level III is concerned, essentially, with the work of the decision aiding analyst. Level IV requires that the participants in the decision aiding process are directly involved, so that they can verbalise key elements of their preferences within the ambit of choices in Level III.

Even if to and fro movements are possible, the order of the construction of a decision aid corresponds to the development of the four levels. The development starts with potential actions. The designated problem is considered by its solutions then by the grassroots participants' declared criteria and preferences. No formal and independent representation of the problem supports MCMDA.

DAP methodology (figure 1) structures the development of a decision aiding framework according to the phases of procedural rationality, *viz*. Intelligence, Design, Choice, Review [Simon1977].

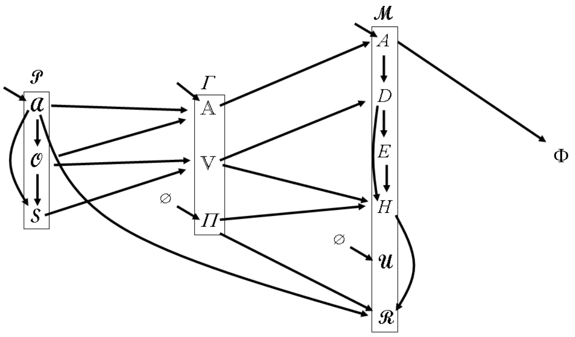


Figure 1 – Artifacts of a Decision Aiding Process approach

This is based on the following hypothesis:

1. The client and the decision-maker are indistinguishable,
2. The client's problem is sufficiently valid and meaningful for the decision aiding analyst and that the client believes in the usefulness of a decision aid.
3. The aid represents a new object that does not seek to change the client's existing situation.
4. The decision process crystallizes the client's problem. The decision aid is considered as a problem resolution. In this sense, decision aiding can be seen as a decision based on a mathematical model.
5. The entry point into DAP occurs *via* an identified problem situation that leads to the development of a final recommendation. This entry point is based on the verbalisations of grassroots participants.
6. Decision aiding is based on models taken from decision theories. The Analyst's role is to prepare the development of these models.

The problem situation seeks to clarify key elements that lead to the problem. It seeks to crystallize the implications for the client and the participants involved in a decision, as response to a problem. This crystallisation also clarifies for the Analyst on which points to focus a decision aid. Formally speaking, the problem situation $\mathcal{P}$ is a triplet ($\mathcal{A}$, $\mathcal{O}$, $\mathcal{S}$) where:

- $\mathcal{A}$ is a set of participants in the decision process.
- $\mathcal{O}$ is a set of stakes of participants that brought them to the decision process.
- $\mathcal{S}$ is a set of engagements taken by participants about their priorities and those of others.

The formulation of the problem aims at developing a response to a problem that is clarified within the scope of a designated problem situation. The formulating of a problem is centered on choices based on the client's rationality in response to an identified problem. It is the task of the Analyst to explicate these choices into a formal representation. These choices are indispensable elements prior to the application of a decision aiding framework. The problem formulation Γ is a triplet [$\mathbb{A}$, $\mathbb{V}$, Π) where:

- $\mathbb{A}$ is the set of potential actions within the framework of problem situation $\mathcal{P}$.
- $\mathbb{V}$ is the set of viewpoints that observes, analyses, evaluates and compares potential actions.
- Π is the decision problem statement – the application typology that considers $\mathbb{A}$ in anticipation of what the client expects.

Based on a numerical representation, that has mathematical properties, the Evaluation model assesses in detail the impact of a projected solution in the formulation of the problem. The

Evaluation model $\mathcal{M}$ is an nth (*A, D, E, H,* $\mathcal{U}$, $\mathcal{R}$) where:

- *A* is the set of alternatives concerning the evaluation model.
- *D* is the set of dimensions, having possible structural properties, with which potential actions *A* are taken into account by the model.
- *E* is the set of scales associated with each element *D*.
- *H* is the set of criteria with which the elements of *A* are evaluated in terms of the client's preferences and the limits of each criteria.
- $\mathcal{U}$ is the set of incertitude distributions associated with *D* and/or *H*.
- $\mathcal{R}$ is the set of information synthesis operators of elements from *A* or of *A*×*A*, namely aggregative operators.

The final recommendation (denoted by Φ) is focussed on the constraints of grassroots reality by a coherence-focussed interrelation of the results of the Evaluation model in terms of a language that makes sense to the client. This involves three basic issues:

1. *technically soundness*: The capacity of the final recommendation to generating an appropriate response to the client's preoccupation.
2. *Operational completeness*: The capacity of the recommendation to be put into practice in a given concrete reality.
3. *Legitimacy*: There is a clear coherence between the recommendation and the context of the decision that may not be (totally) taken into account in the mathematical Evaluation model.

Evaluation models

DAP approach is constructive. Its major strength is that it includes a representation of a problem independent from its solutions. In this context Figure 1 (see above) highlights:

- Based on the model of recommendations, the definition of $\mathcal{R}$ is not directly linked to grassroots participants' preoccupations.
- Π and $\mathcal{U}$ are introduced without reference to the designated problem.

The validation of the process only concerns the $\mathcal{M}$ and thus, the coherence between Γ and Φ. The target reality situation is taken into account in an overall way in creating the Final recommendation.

The introduction of a problem-based evaluation model as an intermediary representation between $\mathcal{P}$ and Γ can reinforce the development possibilities and the capacity of decision aiding (figure 2) by:

- establishing a balance in the shared representations between the client and the analyst in order to better structure their exchanges,
- anticipating the search for preferential-based information linked to the client's preoccupation by interrelating relational artefacts $\boldsymbol{S}$, Π and $\mathcal{R}$,
- increasing possible definitions of choices to the decision aid and an advanced validation,
- constructing initial argued recommendations concerning the structure of the problem,
- creating an adapted model $\mathcal{M}$, (called now the Recommendation Model and denoted by $\mathcal{M}_{\Phi}$),
- a better validation obtained by duplication of links between $\mathcal{P}$ and Γ as well as between $\mathcal{P}$ and $\mathcal{M}_{\Phi}$.

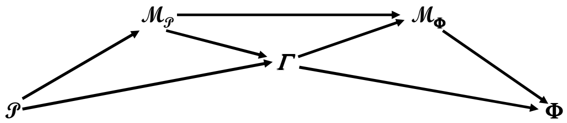


Figure 2 – Problem-based evaluation model

In re-equilibrating the relationship between the client and the analyst, $\mathcal{M}_{\mathcal{P}}$ delimits the decision aid. In presenting a representation of the reality situation as the origin of the aid, $\mathcal{M}_{\mathcal{P}}$ interrelates decision and aid as well as prepares the elaboration of recommendations. The model $\mathcal{M}_{\mathcal{P}}$ also offers distinguishes between a client and a decision-maker, even if they represent the same person in order to question the client's beliefs. From a descriptive approach, $\mathcal{M}_{\mathcal{P}}$ facilitates a vision of decision aiding as an improvement of existing decisions.

Problem evaluation model artefacts

The question of definitions that is independent of notions about decision and decision aiding are found in [Colorni2012]. A decision problem is exanimate trough three key concepts: *values*, *opinions* and *likelihoods*. Values are considered as collective factors, opinions as seen as individual factors, and likelihoods are defined as future preferences conditions that affect a decision. In this context, a decision problem is seen as a sequence of aggregative preferences that is combined arbitrarily to hierarchies of values, opinions and likelihoods. A decision problem constitutes a common element to evaluate $\mathcal{P}$ and build Γ. However note that values and opinions refer to individual and collective factors taken into account by participants to judge the reality. Likelihoods refer to reality as different situations independent of any decision. To express this explicit reference to reality, we prefer use the term verisimilitude. $\mathcal{M}_{\mathcal{P}}$ can thus be represented as a set of values, opinions, verisimilitudes and aggregations that allow to analyze or create coherences into $\mathcal{P}$, as an intention, and Γ, as a future action, or between $\mathcal{P}$ and Γ (figure 3).

As scenarios of future preferences, verisimilitudes make up the imagined structures of participants $\mathcal{A}$. These participants are at the basis of the scenarios that lead to potential actions $\mathbb{A}$, constituting future projects. The common element here is the projective and temporal features of the elements under question. This projection is within the exclusive range of the participants. The ensuing verisimilitudes are designated as Λ.

As mentioned above, the values and opinions represent collective or individual attributes of reality concerned in the decision. The stakes $\mathcal{O}$ are at the basis of these values and opinions. The overall values and opinions lead to viewpoints $\mathbb{V}$. The set bringing together the values O_c and opinions O_i is denoted by O.

The aggregative sequence and the hierarchies represent relations that link values, opinions and verisimilitudes. The engagements $\mathcal{S}$ involve a set of relations between participants and stakes. The problem statement Π interrelates subsets of potential actions. Then aggregative sequence, engagements and problem statement are structures interrelating other artifacts into each triplet $\mathcal{M}_{\mathcal{P}}$, $\mathcal{P}$ and Γ. The overall set of aggregative elements and hierarchies are noted as α.

The definition of a decision problem provides a larger framework to express more than declared preferences. For this reason, we choose to call the building of $\mathcal{M}_{\mathcal{P}}$ as decision problem elicitation.

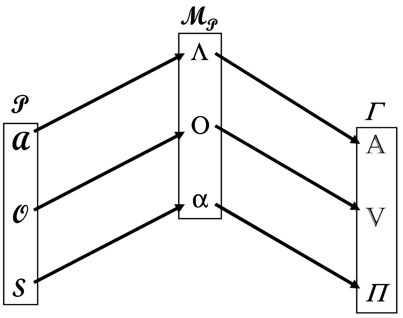


Figure 3 – Links between $\mathcal{P}$, Γ and $\mathcal{M}_{\mathcal{P}}$

The problem evaluation model $\mathcal{M}_{\mathcal{P}}$, comprises the following triplet (Λ, O, α):

- Λ is the set of verisimilitudes
- $O = O_c \cup O_i$ is the set of values O_c and opinions O_i,
- α is the set of aggregations and hierarchies of values, opinions and verisimilitudes.

There is some likeness with the Beliefs-Desires-Intentions concept from the context of multi-agent systems [Wooldridge2002]. Beliefs can be viewed as values and opinions O, desires as verisimilitudes Λ and intentions as engagement $\mathcal{S}$. This framework is useful in group decision support systems. Decision aiding is distinguishable through two central concepts: evaluation and preferences. The set α represents preferences from the decision problem as relations between values, opinions and verisimilitudes. Then the engagements considered in decision aiding are a set of several relations between couples of participants. Intentions consider each agent independently of the others. $\mathcal{M}_{\mathcal{P}}$ aims at evaluating the problem situation $\mathcal{P}$ to build or rebuild the engagements in order to increase the coherence with the problem formulation Γ.

$\mathcal{M}_{\mathcal{P}}$ links Π to $\mathcal{S}$ in $\mathcal{P}$. $\mathcal{S}$ interrelates the elements of $\mathcal{O}$ and $\mathcal{A}$. However Π links up only elements associated to $\mathbb{A}$. Problem statements on $\mathbb{V}$ would allow taking into account hierarchies of values and opinions. The model $\mathcal{M}_{\mathcal{P}}$ enriches the construction of $\mathcal{M}_{\Phi}$:

- $\mathcal{U}$ in $\mathcal{M}_{\Phi}$ does not come from the origin of the problem. However verisimilitudes Λ allow taking into account the participants' decisional behavior, identifying relevant robust scenarios, and clarifying ambiguous situations by constraints or relaxations of values and opinions.
- The operational meaning $\mathcal{R}$ is a delicate question. The need for the independence of each criterion implies that the coherence between the preference model and reality is not self-evident. The identification of dependencies is possible in analyzing the verisimilitudes Λ. The representation of dependencies between criteria in the problem is a question treated in several reflections, as [Marichal2009]. The importance of operational meaning, however, extends from criteria to sets of criteria and this increases the difficulty of the task at hand.
- The control of the combinatorial structure of the problem, in which is found the hierarchies, can be destabilised by a difficult control of the combinatorial explosion in $\mathcal{M}_{\Phi}$. Several elements of $\mathcal{M}_{\mathcal{P}}$ can provide controlling factors. The values O_c can become combinations of dimensions that limit the value domains. The opinions O_i allow the constructions of synthetic, and more satisfying, criteria.

Decision grid

The preference elicitation process brings into play a table of performances to support interactions between the decision-maker and the analyst in order to construct the relationship between potential actions and between criteria [Mousseau2003]. In this context, a matrix-like representation appears as appropriate for $\mathcal{M}_{\mathcal{P}}$ (figure 4).

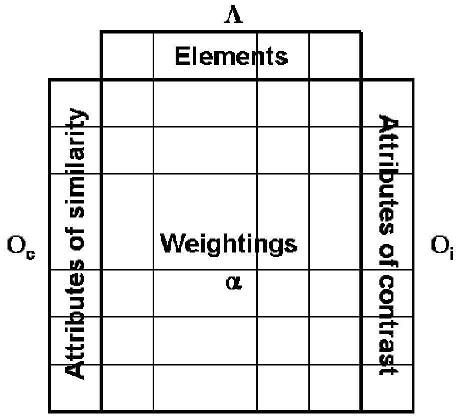


Figure 4 – $\mathcal{M}_{\mathcal{P}}$ as matrix-like decision grid

The creation of the three stages of this matrix is based on the analysis of decision grids – based on the theory of "personal construct" of psychologist George Kelly [Kelly1963]:

1. The grid's method of triads allows the analyst, as an interviewer, to elicit from an interviewee what two meaningful elements (vertical column of the grid), *e.g.* elements A and B, have in common that a third element (Element C) do not have. To this end, the interviewee synthesizes his/her thoughts in one word after discussing it with the decision analyst, for whom the precise meaning of the word must be perfectly clear. The word is then noted to the left of the grid as a similarity attribute, one under another in a list form. This process is repeated until saturation, *i.e.*

when the interviewee has nothing left to say about the elements. The triad of elements is then reordered – *e.g.* what do elements A and C have in common that element B does not have. This is done until all the elements are systematically cross-analyzed. These elements represent the different projects or potential action of actual, realistic or dummy scenarios as the set of verisimilitudes Λ (figure 4)

2. For each attribute of similarity, the interviewee is asked to state what he/she considers as a meaningful contrast to it. The analyst ensures that the interviewee's declared contrast is not an unreflected antonym. When a satisfactory term is found, it is duly noted down to the right of the grid. Obtained by the triadic principle the attributes of similarity represents the collective expression of the elements for the interviewee's. Each attribute of contrast refer to a lonely attribute of similarity and then express a personal expression of the elements. According to a process of continued socialisation [Elias1991] the attributes of similarity represents the values O_c and the attributes of contrast the opinions O_i (figure 4).

3. A five-point numeric weighting transforms the interviewee's vision into an aggregation of hierarchies of values, opinions and verisimilitudes. The five-point scale avoids a too wide dispersion for the interviewee to handle. The attribution of the weightings is done where:

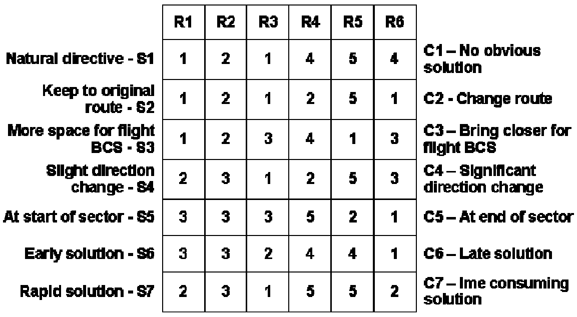
- "1" applies if the ascribed attribute is seen as *very close* to the similarity pole (left side of the grid) and "5" if the attribute *very close* to the contrast pole (right side of the grid).
- "2" applies if the attribute is seen as *more or less* close to the similarity pole, and "4" if the attribute is *more or less* close to the contrast pole.
- "3" applies is the element is not close to either poles.

These weightings allow to represent the hierarchies of the decision problem as several relations between attributes of values and opinions, between scenarios of verisimilitudes and between attributes and scenarios (figure 4).

In this way the weighting, and its accompanying commentaries, establishes a meaningful subjective evaluation process [Grabisch2009] where the specificities of values and opinions can be clarified. This numeric representation of the decision problem expressed by the decision maker, called decision grid, shows the gaps in performances and preferences, between attributes, between elements, as well as between subsets of attributes and subsets of elements. The grid is grounded in the *Dominance-based Rough Set Approach* [Greco2001].

Illustration

The decision grid was used in the context of air traffic control. To build a decision aid, the decision grid allows identify the system of value and the preferences structures the air traffic controllers use in their decisions [Annebicque2012]. In this case, six possible resolutions to avoid a crash between two aircrafts are depicted as six elements (R1 to R6). An interview with an air traffic controller produces seven similarity (S1 to S7) and contrast attributes (C1 to C7) (figure 5).



	R1	R2	R3	R4	R5	R6	
Natural directive - S1	1	2	1	4	5	4	C1 – No obvious solution
Keep to original route - S2	1	2	1	2	5	1	C2 - Change route
More space for flight BCS - S3	1	2	3	4	1	3	C3 – Bring closer for flight BCS
Slight direction change - S4	2	3	1	2	5	3	C4 – Significant direction change
At start of sector - S5	3	3	3	5	2	1	C5 – At end of sector
Early solution - S6	3	3	2	4	4	1	C6 – Late solution
Rapid solution - S7	2	3	1	5	5	2	C7 – Ime consuming solution

Figure 5 – Conflict resolution decision grid

The system of values is decomposed into three groups. The first group concerns issues of regularity regarding the specificities of a resolution (S_1, S_2 and S_4). The other two groups focus on safety problems as the airspace available for

resolution (S_5, S_6 and S_7) and the general traffic around the conflict (S_3).
The weighting system highlights three resolutions groups. These are the natural (R_1, R_2, R_3 weighted by 1, 2 or 3 only), degraded (R_4 and R_5 weighted by 4 and 5), and new (R_6 weighted one time by 4 only) resolutions.

Due to a lack of space, only the natural resolutions are analyzed in detail here. When the grid is limited to the sub-group of “natural” resolutions, relations between attributes shows several ones are not differentiable. For example, attributes S_4 and S_7 differentiate R_1, R_2 and R_3 in the same way and S_5 does not differentiate. So S_1, S_3, S_4 and S_6 appear significant only. In a preference structure (I, P, Q, R) with:

$p_{j,k} - p_{i,k} = 0 \Rightarrow R_i \, I_{Sk} \, R_j$

$p_{j,k} - p_{i,k} = 1 \Rightarrow R_i \, Q_{Sk} \, R_j$

$p_{j,k} - p_{i,k} \geq 2 \Rightarrow R_i \, P_{Sk} \, R_j$

where R_n is an element in the column n, S_m is the attribute on the line m and $p_{n,m}$ is the weighting at the intersection of the column n and the line m, the system of preference can thus describe the proximity of values in the following way:

$R_1 \, Q \, R_2$

$R_3 \, I_{S1} \, R_1$, $R_3 \, Q_C \, R_1$ with C = {S_4, S_6} and $R_1 \, P_{S3} \, R_3 \Rightarrow R_1 \, R \, R_3$

$R_3 \, P_{S4} \, R_2$, $R_3 \, Q_C \, R_2$ with C = {S_1, S_6} and $R_2 \, Q_{S3} \, R_3 \Rightarrow R_2 \, R \, R_3$

The comparison between R_1 and R_2 is clear for all the declared attributes. We can note that:
- S_3 plays a determining role in the incomparability of R_1 and R_2 with R_3.
- S_1 and S_6 do not play a determining role in the comparison of R_3 with R_1 and R_2 as they accord a weak preference at R_3.
- S_4 accords a strict preference for R_3.

The conflict among criteria is therefore focused essentially on S_3 and S_4 and they belong to two groups with different values. S_3 refers to safety issues as all conflict (air crash) resolutions must ensure not to generate a conflict that did not exist before. A conflict resolution concerning an aircraft, distinct from the rest of the traffic, thus needs close supervision. S_4 concerns traffic regularity and becomes relevant when a conflict resolution is in progress. For these reasons, it is possible to accord to S_3 a right of veto if it is accorded a strong preference. The preference is thus accorded to R_1 rather than to R_3 and to R_3 rather than R_2.
An examination of S_4 and S_6 brings out significant information for the evaluation of the order of conflict resolution and the model $\mathcal{M}_\Phi$. The relations:

$R_1 \, Q_{S4} \, R_2$

$R_1 \, I_{S6} \, R_2$

show that the aircraft course heading and moment of application are disassociated. They can therefore constitute two independent criteria. The relations:

$R_3 \, Q_{S4} \, R_1$

$R_3 \, P_{S4} \, R_2$

$R_3 \, Q_{S6} \, R_1$

$R_3 \, Q_{S6} \, R_2$

show that R_3 is a more precise resolution and that the cape orders are discrete. This information provides the possibility to reduce the combinatorial structure coming from the value domains. These observations underline that the handling of an air crash conflict does not seek out the optimal solution but rather to satisfy a constraint. This is a vital factor in the choice of a problem statement and the aggregative model.

A similar representation for R_4, R_5 and R_6 is not analysed here. For R_4 and R_5, the grid highlights the impact of a degradation of parameters involved in a natural resolution on the system of values. This type of resolution can be applied by air traffic controllers when the air traffic situation is complex or the values insufficient and needs to be supplemented by a controller’s professional “opinions” as weightings 4 and 5 show. Then the robustness of a decision aid can be obtained by limiting its intervention to natural resolutions in accordance to values and in leaving a degraded situation in the skilled hands of an air traffic controller.

Regarding the weightings R_6 is to R_1, R_2 and R_3, appears coherent with the declared values. The proximity to the contrast attribute C_1 shows an action that is not yet put in place but exists on the level of considered opinions.

Conclusion

The example of air traffic control highlights the relevance of the Evaluation model $\mathcal{M}_{\mathcal{P}}$ in the context of descriptive decision aiding. In doing this, the Model facilitates an examination of the reality of a decision and the identification of its limits. A decision aiding framework can thus be validated in order to identify a relevant aid, or a combination of aids.

Our Evaluation model $\mathcal{M}_{\mathcal{P}}$ provides a highly useful tool for the decision analyst in structuring exchanges with the client. The Model is based on a Decision Aiding Process that provides a coherent methodological framework grounded on known concepts in decision aiding. The Model $\mathcal{M}_{\mathcal{P}}$ represents a useful tool in a constructive approach to decision aiding. Further examinations in this domain and in exploring related experiences in decision aiding will allow drawing richer lessons.

Bibliography

[Annebicque2012] Annebicque D., Crévits I., Poulain T., Debernard S., Millot P. « Decision Support Systems for Air Traffic Controllers based on the Analysis of their Decision-Making Processes » Int. J. Advanced Operations Management, vol. 4, nos. 1/2, pp. 85-104.

[Colorni2012] Colorni, A., Tsoukiàs, A. « What is a decision problem? ». Séminaire du LAMSADE, 25 janvier 2012.

[Elias1991] Elias, N. « The Symbol Theory » Sage, London.

[Grabisch2009] Grabisch M. « Subjective evaluation » in Decision-Making Process: Concepts and Methods. Wiley.

[Greco2001] Greco, S., Matarazzo, B., Slowinski, R. « Rough sets theory for multicriteria decision analysis » European Journal of Operational Research 129, 1-47.

[Kelly1963] Kelly G.A. « The Psychology of Personal Constructs » Norton, New York.

[Marichal2009] « Aggregation functions for decision making » in Decision-Making Process: Concepts and Methods. Wiley.

[Mousseau2003] Mousseau, V. « Elicitation des préférences par apprentissage constructif », Habilitation à Diriger des Recherches, Université Paris-Dauphine.

[Roy1985] Roy, B « Méthodologie Multicritère d'aide à la décision » Economica.

[Simon1977] Simon H.A. « The new science of management decision » Prentice Hall.

[Tsoukiàs2007] Tsoukiàs, A. « On the concept of decision aiding process » Annals of Operations Research 154, 3-27.

Group Decision Making for selection of an Information System in a Business Context

Teresa Pereira
Escola Superior de Estudos Industriais e de Gestão do Politécnico do Porto
Rua D. Sancho I 981, 4480-876 Vila do Conde, Portugal
Email: teresapereira@eu.ipp.pt
Dalila B. M. M. Fontes
Faculdade de Economia da Universidade do Porto, and LIAAD - INESC-TEC L.A.
Rua Dr. Roberto Frias, 4200-464 Porto, Portugal
Email: fontes@fep.up.pt

Abstract

The main objective of this work is to report on the development of a multi-criteria methodology to support the assessment and selection of an Information System (IS) framework in a business context. The objective is to select a technological partner that provides the engine to be the basis for the development of a customized application for shrinkage reduction on the supply chains management. Furthermore, the proposed methodology differs from most of the ones previously proposed in the sense that 1) it provides the decision makers with a set of pre-defined criteria along with their description and suggestions on how to measure them and 2) it uses a continuous scale with two reference levels and thus no normalization of the valuations is required. The methodology here proposed is has been designed to be easy to understand and use, without a specific support of a decision making analyst.

Key words: Group decision, multi-criteria method, information systems.

1 Introduction

The development and use of Information Systems (IS) has been actively pursued by organizations for maintaining their competitive advantages in today's dynamic environment. The assessment and selection of IS applications is complex and challenging, since it often involves (a) multiple decision makers, (b) multiple selection criteria, and (c) subjective and imprecise assessments. To ensure that the best possible IS is selected with proper justification, it is desirable to use a structured approach capable of comprehensively analyzing the performance of available IS in a specific decision setting.

The group decision-making process is very difficult since it involves the presence of multiple decision-makers each of which has his/her own perception on how the problem should be addressed and on how the decision process should be guided [10]. Therefore, when multiple actors participate in a decision, it is necessary to aggregate their opinions, which can be made apriori, i.e., the group acts together as a unit, or aposteriori, i.e., aggregating the individual opinions by using some sort of priorities, see e.g. [7]. A discussion and review on these methods and their application to specific problems can be found in [11]. In here, we propose an aggregation process that although based on a group consensus, it starts by analyzing individual preferences. In our case the Decision Makers (DMs) must agree on the evaluation given to each criteria as well as on the weights that are to be associated with the criteria. However, they start by performing individual evaluations. By doing so, they have to justify to each other their opinion and thus discussion is forced.

While multi-criteria methods are well known and many different applications have been reported, apparently with exception of AHP, they are not often used in the field of IS selection due the huge number of criteria that need to be assessed and also due to teh existence of imperfect information, see [4].

they are not often used in the field of IS selection. The method, however, can be used to assess the relative attractiveness of alternative ways of accomplishing virtually any specified ends. For instance, in [14] an Enterprize Resource Planning (ERP) implementation framework has been proposed as a guide for small manufacturing enterprizes considering ERP implementation. This framework integrates simulation and can be used to better meet the goals of reducing implementation costs while increasing desired achievement levels. Multi-criteria studies in finance and accounting problems such as bankruptcy prediction, mergers and acquisitions, auditing, share repurchases can be found in [1] and the references therein. For recent surveys on multi-criteria applications see [2, 6].

Here, we propose a Multi-criteria Decision Aid (MCDA) methodology, which can be characterized as a non-linear and recursive process to select an option among several. The methodology does not aim at finding the best decision, but rather to guide the Decision Maker (DM) through the process of selecting one that best suits their goal and their understanding of the problem. Given that a solution is characterized by many different criteria, usually there is no single solution that performs better for all criteria. In addition, the existence of several DMs makes it even harder, if not impossible, to find a solution which is better for all of them. Thus, tools aiding in the decision making process are needed in order to force discussion, objectivity, and quantification. However, many of the tools available to DMs are not easy to use and require the presence of an analyst to lead the process.

The methodology we propose here is simple to use and requires a small effort to understand and use it. It has been tested on a real application to single global decision regarding the selection of a IS, as reported in Section 3. The DMs were able to perform the final evaluation and to reach a decision by themselves, i.e. without an analyst. Therefore, our contributions are twofold. On the one hand, we address the IS selection problem, which has not been addressed before. On the other hand, our methodology differs from the previously proposed ones in the sense that it uses a continuous scale with seven semantic levels with two reference and thus when quantified no normalization is required. Furthermore, it provides additional help to the GDM since it provides an original set of criteria, that can refined by GDM by removing, or modifying, or adding new criteria, along with their description and suggestions on how to measure them.

The rest of the paper is organized as follows. To begin with, in Section 2 we explain the multi-criteria methodology proposed. Then, in Section 3 we present the background of the decision situation, i.e., a case study. Finally, in Section 4 some conclusions are drawn and a discussion of future work is provided.

2 MMASSITI Methodology

A group of DMs faces the problem of choosing one alternative, over all possible others. In order to do so, the DMs must first identify the set of criteria to be used in the analysis of the the alternative solutions, i.e. what will be used to measure desirability or attractiveness of the alternatives. In the methodology we propose this is done in two phases, see Figure 1. In phase 1, the DMs determine a set of requirements, all qualitative in nature, that the available IS must satisfy in order to be considered as a possible decision alternative. Therefore, at the end of phase 1, a reduced set of alternative decisions, to be analyzed further in phase 2, have been identified. However, if the set of alternatives is though to be too large, further analysis may be performed in order too reduce it. Furthermore, the first phase is also intended to help the DMs to structure the problem, since it helps them to think about the IS assessment and its alignment with the organization's strategies and existing resources. Then, in phase 2 the DMs must specify the criteria to be used to evaluate the IS alternatives, i.e. technical requirements, functionalities, reliability, costs, customization, implementation time, etc.. These criteria include both quantitative and qualitative aspects. In this phase, the DMs must also define the weights to be used to obtain a global evaluation for each alternative through aggregation. Then, before presenting the better alternative, according to the criteria chosen and the evaluations provided by the DMs, robustness and sensitivity analyses are performed.

Significant research has been produced in the

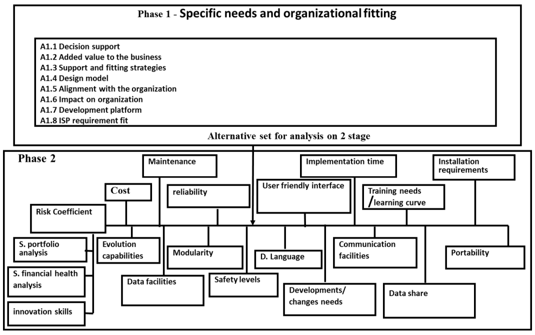

Phase 1 - Specific needs and organizational fitting
A1.1 Decision support
A1.2 Added value to the business
A1.3 Support and fitting strategies
A1.4 Design model
A1.5 Alignment with the organization
A1.6 Impact on organization
A1.7 Development platform
A1.8 ISP requirement fit
Alternative set for analysis on 2 stage
Phase 2
Maintenance
Implementation time
Installation requirements
Cost
reliability
User friendly interface
Training needs /learning curve
Risk Coefficient
S. portfolio analysis
Evolution capabilities
Modularity
D. Language
Communication facilities
Portability
S. financial health analysis
Data facilities
Safety levels
Developments/ changes needs
Data share
innovation skills

Figure 1: Structure of the proposed methodology.

multi-criteria decision area proposing several multi-criteria methodologies and applications. Some authors, such as Vincke [15], divide the methods in American aspiration and European aspiration. Regarding the American aspiration ones, the most popular and widely used are AHP - Analytic Hierarchy Process [13] and MAUT/MAUVT - Multiple Attribute Utility/Value Theory. The AHP decomposes the original problem into sub-problems that can be solved independently. Its popularity is mainly due its software support - Expert Choice-which uses pairwise comparisons along with a semantic and ratio scale to assess the DM preferences. This hierarchical model is useful in many situations; however, it is not easy to apply because of its axiomatic foundations. It assumes that there must be outer and inner independence between the different hierarchical levels and elements, which is not always easy to verify, as is the case of IS for business context. In what concerns MAUT, the most popular applications are SMART methods - simple multi-attribute ranking technique [8] and SMARTER (SMART Exploiting Ranks), an extended version due to Edwards and Barron [5]. In these cases, the different points of view are aggregated into a unique function that must subsequently be optimized. UTA - Utility Additive Method is an indirect method of applying MAUT, through PRECALC, an interactive software [9]. Within the European methods we can find ELECTRE - ELimination Et Choix Traduisant la REalité (ELimination and Choice Expressing REality) [12] and PROMETHEE - Preference Ranking Organization METHod for Enrichment Evaluation [3]. The former comprises two main parts: the construction, which compares each pair of actions and the exploitation, which provides recommendations based on the results previously obtained. Many applications are reported in chapter 5 of [6]. PROMETHEE, which is also based on pairwise comparisons (as is the case of ELECTRE) has successfully been used in many decision making contexts worldwide, for a non-exhaustive list see [2].

MMASSITI is a multi-criteria methodology for assessing and selecting information system and it has been designed to be easy to understand and use, without a specific support of a decision making analyst, to offer the Group of DMs (GDM) an effective support decision-making process, and to act as enhancer of the specification accuracy. The methodology intends to be simple so that the GDM can be lead through it considering the following steps:

Step 1: Define the consistent and coherent family of criteria in consensus;

Step 2: Analyze and validate the description of each criterion and define how to measure it;

Step 3: Define the requirements and requirement levels for the reference levels "neutral" and "better" (these requirements may be adjusted later, when evaluating alternatives);

Step 4: Establish the relative importance weights to be associated to each criterion;

Step 5: Find out the largest value for the seven semantic levels. (S_3^-: Much Worst, S_2^-: Worst, S_1^-: Slightly Worst, S_0: Neutral, S_1^+: Slightly Better, S_2^+: Better and S_3^+: Much Better).

Step 6: Assess each alternative on each criterion and assign a collective value in accordance with the range defined in step5.

Step 7: Compute the aggregated global score for each alternative, using the additive model

Step 8: Sensitivity and robustness analyses;

2.1 Defining and evaluating criterion

In our methodology, the GDM is presented with a set of pre-defined criteria that does not address a specific IS, but rather generally covers all the criteria, taking into account the choice of any IS in an organizational context. The intention is to present to the GDM a "starting point". Nevertheless, it is the GDM that defines and validates the consistent and coherent family of criteria by restricting, or modifying, or adding new criteria to initial family of criteria they were presented with.

Our multi-criteria methodology uses a continuous scale, rather than the usually used discrete scale, with seven semantic levels. For each of these levels the GDM finds a maximum numerical value within $[-100, 100]$, except for "Neutral" that is valued as 0 (S_{-3}: Much Worst, S_{-2}: Worst, S_{-3}: Slightly Worst, S_0: Neutral, S_{+1}: Slightly Better, S_{+2}: Better, and S_{+3}: Much Better). Then, an interval of possible evaluations or each semantic level is computed by using the following relation $S_{j-1} <= S_j <= S_{j+1}$. It should be noticed that no scale values is required. The range of values for each of the semantic levels remains the same throughout the whole decision making process (regardless of the criterion or of the alternative under evaluation).

Each DM decides, individually and independently, on which value of the semantic scale to put each criterion for each alternative. Then, a discussion follows among the DMs in order to find a consensus final scale (for each criterion and each alternative). Then, for each criterion and each alternative the decision makers, individually, provide a range of values within the range previously defined for the semantic scale. With these ranges a common range is found and the DMs are provided with it as well as with its median value. The GDM must then find a consensus value x_i^a for each criterion on each each alternative, which may or may not be the suggested one (the median), however it has to fall in the common range.

2.2 Computing weights and the aggregated global value

The swing weight procedure [17] is used for finding out the weight value v_i for each criterion. These values must be obtained by a consensus amongst the GDM. The collective relative value of each criterion is defined in relation to the most important one, which has a value of 100. Once all weights have been found, their value is normalized using the Weber and Borcherding formulae [16]:

$$w_i = \frac{v_i}{v}, \text{ where } \sum_i v_i.$$

The aggregated value of each alternative $x(a)$, is obtained by aggregating the utility value of each alternative on each criterion x_i^a. In order to do so, we use the additive model due to its simplicity and transparency (to the GDM).

$$x(a) = \sum_i w_i x_i^a.$$

2.3 Sensitivity and robustness analyses

Sensitive and robustness analyses are important to assure GDM confidence on the methodology results. In the sensitive analysis we evaluate the impact of the variation of the weight of a criterion using the full range of the scale. For this specific work we propose to recompute the aggregated values for all alternatives considering the following 6 scenarios.

1. the weights in the second phase are all considered equal, while in the first phase their value remains unchanged;

2. the weights have all the same value in both phases;

3. vary the value of one criterion at a the time in their full range;

4. vary the value of the two most important criteria at the same time, while the rest remain unchanged;

5. vary the value of the three most important criteria at the same time, while the rest remain unchanged.

Regarding the robustness analysis, each criterion value is varied, one at the time. Several values for the weights are considered and the range of the variation is bounded, since the criteria relative order cannot be affected.

3 Case Study

The methodology was tested on a retail software company. The company wishes to select a technological partner to supply an engine that will be used as a basis for the development of a customized application for shrinkage reduction on the supply chains management.

In a first stage, meetings were held in order to introduce and explain the methodology to the project management team, which was composed of the: Shrinkage reduction business expert; OnLine Analytical Processing IT expert; Data warehousing expert; Decision Support systems expert; Product manager.

To perform the first phase of the methodology, business and technical requirements were specified and readjusted by the GDM, based on the suggested family of criteria, see Table 1. The family of criteria proposed to the GDM was reached by performing theoretical search, based on the literature, and empirical work, questionnaires have been sent to 300 companies and interviews have been conducted with 14 companies. Several meetings were held before a finally set of criteria has been agreed upon.

Having defined the requirements that the available IS must satisfy to be an alternative, a market search was conducted. Five alternative IS were found to be of interest and thus their merits are to be analyzed. The alternatives in analysis will be referred to in this paper as A, B, C, D and E.

Using the same methodology of phase 1, in phase 2 we have reached the criteria given in Table 2. These are the criteria on which each alternative IS is going to be evaluated.

The next stage, involved setting up the relative importance ranking (weight) assigned to each criterion according to the swing weight procedure [17]. Once this was achieved, the normalized criterion weights were computed, as in Section 2.2. Next, it follows the evaluation of each alternative on each criterion. In order to do so, and as explained before in Section 2.1 a fixed scale with seven levels was used, two of which are reference values.

The company though that a full evaluation of the 5 alternatives would be a very costly process. Therefore, an evaluation using the criteria of phase 1 was performed in order to find out the which were the best alternative, that should be chosen for further evaluation. The weighted additive aggregation, see Table 3, shows that the alternatives A and B both scored 48.22, while the remaining alternatives all have similar scores and scored a little less than alternatives A and B. (C scored 44.55, D scored 44.02, and E scored 42.36), all with very similar scores.

The methodological procedure in phase 2 is similar to that of phase 1, but now considering phase 2 criteria, as given in Table 2. In addition, the global score of phase 1 is used in phase 2, since the global aggregated value of each alternative on the A1 criterion automatically goes to phase 2. As already said, the GDMs have decided that only the two best alternatives are to be analyzed in phase 2. The consensus evaluation obtained is given in Table 4.

As it can be seen in Table 4, alternative A has the best score, with a aggregated global value of 49.45, while alternative by B has a aggregated global value of 47.15. These values are quite similar, which was not a surprise since the alternatives have similar functionalities. Nevertheless, sensitivity and robustness analyses were carried out with several sets of scenarios, as explained in Section 2.3, and the order has always remained the same.

It should be noticed that, the results are only valid for this analysis scope, this company and this GDMs.

4 Conclusions

The methodology here proposed had as a main objective to be able to be used by decision makers without the presence of experts in multi-criteria decision methodologies. This was achieved since the project management team was able to apply the methodology themselves after meetings were held in order to introduce and explain them the methodology.

Another important issue, in what concerns practical utilization, is the pre-defined set of criteria that the GDM are provided with. This set of criteria was proposed after performing theoretical search, based on the literature, and empirical work, based on questionnaires and interviews. A description of the criteria and guidelines on how to quantify them are also provided. In addition, the proposed methodology has the advantage of not

A.1 Needs of the organization	Results
A1.1 Entrance cost	Aggregated expected value
A1.2 Business added value	Set of alternatives
A1.3 Support/fit company strategies	
A1.4 Platform development	
A1.5 Requirements fit	

Table 1: Criteria used in phase 1, resulting from the company business and strategic plan.

requiring normalization, regarding criteria evaluation, since it uses a continuous scale with two reference levels.

The sensitivity and robustness analyses have shown the methodology to be reliable since the recommendations have remained the same under several different scenarios.

Furthermore, we address the IS selection problem, which has not been addressed before (except for the ERP particular case). A case study has been reported.

Currently we are working on the implementation of the methodology through a decision support system. This will make the methodology even easier to use since all data will be introduced through an user-friendly graphical interface.

References

[1] D. Andriosopoulos, C. Gaganis, F. Pasiouras, and C. Zopounidis. An application of multicriteria decision aid models in the prediction of open market share repurchases. *Omega*, 40(6):882–890.

[2] M. Behzadian, R.B. Kazemzadeh, A. Albadvi, and M. Aghdasi. ROMETHEE: A comprehensive literature review on methodologies and applications. *European Journal of Operational Research*, 200:198—-215, 2010.

[3] J.P. Brans. La méthode promethee. In *L'ingénierie de la décision: élaboration d'instruments d'aide à la décision*, Québec, Canada, 192. Presses de l'Université Laval.

[4] W. Chun-Chin, C. Chen-Fu, and J.W. Mao-Jiun. An AHP-based approach to ERP system selection. *International Journal of Production Economics*, 96:47–62, 2005.

[5] W. Edwards and F.H Barron. SMART and SMARTER: improved simple methods for multiattribute utility measurement. *Organizational Behavior and Human Decision Processes*, 60(1):306–325, 1994.

[6] J. Figueira, V. Mousseau, and B. Roy. *Multiple Criteria Decision Analysis: State of the Art Surveys*, volume 78 of *International Series in Operations Research and Management Science*. Springer-Media, New York, 2005.

[7] E. Forman and K. Peniwati. Aggregation individual judgments and priorities with the analytic hierarchy process. *European Journal of Operational Research*, 108:165–169, 1998.

[8] R. L. Keeney and H. Raifa. *Decisions With Multiple Objectives: Preferences and Value Tradeoffs*. John Wiley & Sons, New York, 1976.

[9] R.L. Keeney. *Value-Focused Thinking: A Path to Creative Decision-Making*. Harvard University Press, 1992.

[10] N. Matsatsinis, E. Grigoroudis, and Samaras A. Aggregation and disaggregation of preferences for collective decision-making. *Group Decision and Negotiation*, 14:217–32, 2005.

[11] D.C. Morais and A.T. Almeida. Group decision making on water resources based on analysis of individual rankings. *Omega*, 40:42–52, 2012.

[12] B. Roy. Classement et choix en présence de points de vue multiples (la méthode ELECTRE). *La Revue d'Informatique et de Recherche Opérationelle (RIRO)*, 8:57–75, 1968.

A2: Vendor Risk Coefficient - IS maturity risk - Financial risk - Costumer portfolio - Innovation skill	Results - Qualitative scale
A3: Licensing and Cost - Licensing and support cost	Results - Qualitative scale
A4: Maintenance - Annual cost - Conditions	Results - Maintenance cost/year
A5: IS perceived reliability - Supplier customer portfolio - System demonstration - IS portfolio - IS life cycle	Results - Qualitative scale
A6: User friendly interface - Excel look and feel	Results - Qualitative scale
A7: Training Needs - Training quality - Training cost	Results - Training quality - technical staff no. and time
A8: Modularity facilities - Types of modularity	Results - Qualitative scale
A9: Evolution capabilities - Open systems -Future developments	Results - Qualitative scale
A10: Development Complexity -Learning curve - Development language	Results - Hours/technical staff times hour cost
A11: Safety levels - Customized - Transition position follow-up	Results - Qualitative scale
A12: Communication features - (WEB; EDI, CIM, CRM, etc.) - Standard protocols	Results - Qualitative scale
A13: Data share - Shared entities - Managing high volume/detailed data	Results - Qualitative scale
A14: Product stability/Support	Results - Qualitative scale
A15: Deployment/Implementation Cost - Estimation schedule - Additional Human Recourses (HR) - Additional IS	Results - Number of hours - HR cost -IS cost

Table 2: Criteria used in phase 2, for evaluating the IS alternatives.

	Criteria A1						
	1.1	1.2	1.3	1.4	1.5	value	
Swing	75	100	70	60	95	255	Global
Weight	0.188	0.25	0.175	0.15	0.238	1	Scale (1–100)
A	-15	0	0	-5	0	-3.57	48.22
B	0	0	0	0	-15	-3.57	48.22
C	-50	0	0	-10	0	-10.9	44.55
D	-15	-5	0	-5	-30	-11.96	44.02
E	-75	0	0	0	-5	-15.29	42.36

Table 3: Evaluation of the criteria in phase 1 and the aggregated global value.

Phase 2			IS Alternatives	
criteria	Swing	Weight	A	B
A1	95	0,081	-3,57	-3,57
A2	80	0,068	0	0
A3	95	0,081	10	0
A4	80	0,068	0	-15
A5	80	0,068	-10	10
A6	100	0,085	0	-15
A7	95	0,081	-15	0
A8	70	0,060	0	0
A9	70	0,060	-10	0
A10	80	0,068	0	-30
A11	50	0,043	0	0
A12	50	0,043	0	0
A13	90	0,077	0	0
A14	70	0,060	-15	0
A15	70	0,060	30	-30
Global	1175	1	-1.1	-5.7
value			(49.45)	(47.15)

Table 4: Evaluation of the criteria in phase 2 and the aggregated global value.

[13] T.L. Saaty. *The Analytic Hierarchy Process.* McGraw-Hill, New York, 1980.

[14] A.Y.T Sun, A. Yazdani, and J.D. Overend. Achievement assessment for enterprise resource planning (ERP) system implementations based on critical success factors (CSFs. *International Journal of Production Economics*, 98:189–203, 2005.

[15] P Vincke. *Multi-criteria Decision-aid.* John Wiley & Sons, New York, 1992.

[16] M. Weber and K. Borcherding. Behavioral influences on weight judgements in multiattribute decision making. *European Journal of Operacional Research*, 67:1–12, 1993.

[17] D.V. Winterfeldt and W. Edwards. *Decision Analysis and Behavioural Research.* New York, 1986.

Ontology-based management of uncertain preferences in user profiles

Joan Borràs[1,2], Aïda Valls[2], Antonio Moreno[2], David Isern[2]

[1] Science & Technology Park for Tourism and Leisure,
C/ JoanotMartorell, 15. 43480 Vila-Seca, Catalonia (Spain)
joan.borras@pct-turisme.cat
[2] Intelligent Technologies for Advanced Knowledge Acquisition (ITAKA)
Departament d'Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili.
Av. Països Catalans, 26. 43007 Tarragona, Catalonia (Spain)
aida.valls@urv.cat, antonio.moreno@urv.cat, david.isern@urv.cat

Abstract. Ontologies define a set of concepts related to a certain domain as well as the relationships among them. This structure may be exploited to represent and reason about the preferences of a user. The user profile stores the degrees of interest of the user on several concepts using membership functions. In this way, each concept in the ontology is a fuzzy set and any user belongs to this fuzzy set to a certain degree. To represent the uncertainty on this information, a degree of confidence on the membership value is also included. After an initial assignment of preferences, spreading algorithms that exploit the taxonomical information of the ontology are applied to propagate the information about the user's preferences (and their associated uncertainty) through the whole set of concepts. This framework for managing uncertain preferences has been successfully applied in a Tourism recommender system[1].

1 Introduction

In the current context of information overload, people are daily confronted with many situations in which a decision must be taken in the presence of a wide set of alternatives defined on a large number of criteria or attributes. *Recommender systems* (RS) can be very helpful in these situations, because they can analyse automatically all the information available on the possible alternatives, compare it with the user preferences or interests, rate the alternatives and present to the user the most appropriate ones. Thus, a basic component of RS is the *user profile*, which stores the preferences.

A current research trend is the design of *semantic recommender systems* (SRS), in which the semantic information about the domain, usually represented in the form of an ontology, is used to represent both the user profile and the recommendable items. As pointed out in [3], SRS provide the benefits of semantic richness (preferences are richer and more detailed than the standard ones based solely on keywords), hierarchical structure (allowing an analysis of preferences at different abstraction levels) and inference (the structure of the ontology may be used to reason about the preferences on all the domain concepts). As will be mentioned in the next section, some authors have already proposed works with ontology-based user profiles, and where the ontology components (especially the concepts and the taxonomic relationships between them) are used to spread preference information through the ontology, to compare users to form clusters of people with similar tastes (in collaborative filtering systems) or to match the user preferences with the representation of each item (in content-based RS). In those systems the user profile is usually built and maintained through explicit information provided by the users (filling forms, rating items) or implicit information related to the interaction of the user with the RS (saving items, deleting items). However, up to our knowledge, the uncertainty associated to these kinds of information has not been appropriately considered and incorporated into the management of the user profile. The work presented in this paper intends to fill this gap, by proposing a general framework that allows representing and reasoning about the uncertainty associated to preferences in ontology-based SRS.

The rest of the paper is structured as follows. Section 2 reviews some related work on SRS, and points out the lack of management of the uncertainty associated to the sources of information. Section 3 explains the representation of preferences and uncertainties in the proposed framework, detailing how preferences can be easily initialized and how this initial information may be propagated downwards the ontology. Once the user has interacted with some recommended activities, the spreading algorithms detailed in section 4 update the preference and uncertainty information on all the ontology concepts. This framework has already been applied to a specific system for recommendation of touristic activities, as described in section 5. The last section provides the final conclusions and outlines some points of future work.

[1] The content of this paper has been already published in the CCIS series of Springer, as Proceedings of the Conference IPMU, 2012 [10]. Copyright corresponds to Springer publishers. http://www.springerlink.com/content/g881611130231446/

2 Related work

Recommender systems require a user profile that stores the degree of interest on each different criterion that describes an item [5]. *Semantic-based recommender systems* use the semantic knowledge stored in a domain ontology to improve the accuracy of the recommendations.

One possibility is to represent both the user preferences and the domain objects using the ontology concepts. Then, the relationships between them may be used to evaluate the similarities between the user interests and the recommendable items. For example, in [1] the user profile includes all the items that have been bought by the user, along with an interest between 0 and 1. This information is transferred to the concepts that are leaves of a domain ontology. In [3] both user preferences and items are represented as a set of weights between -1 and 1 associated to the concepts of an ontology. They also propose the idea of representing user stereotypes in the same way. The initial preferences are spread through the ontology by taking into account different kinds of semantic relationships between concepts. In [7] the initial user profile associates a weight 1 to each ontology concept. By analysing the documents with which the user interacts, the weights of the user profile are dynamically updated. This information is later spread through the ontology, by considering a particular pre-computed relationship weight between each pair of ontology concepts. A collaborative version of the same idea is applied in[8]. In [9] a content-based RS that crawls and clusters scientific papers according to their keywords is presented. The system matches those items with a personal ontology of concepts related to the user.

The use of an ontology to represent user profiles permits their comparison in collaborative filtering systems. As an example, in [4], the profile stores the tags employed by the user in a social network, which belong to a predefined taxonomy. By reasoning on the taxonomical relationships it is possible to compute the semantic similarity between users, and recommend to a user the items that similar users have tagged. In [3] the authors propose to identify communities of interest from the tastes and preferences expressed by users in personal ontology-based profiles. A user receives advertisements about items that have been positively valued by other users in the same cluster. Collaborative filtering using ontology-based user profiles is also applied in [6]. In this case, the authors propose to connect user profiles creating a social folksonomy and to provide a user with a recommendation of similar users in the network.

It is worth noting that, in all the works that represent user profiles through ontologies, the explicit management of uncertainty has not been considered (neither in the user profile representation nor in the propagation of this information through the ontology). The main novel component of the work proposed in this paper is the careful consideration of the explicit and implicit sources of information about the user preferences in order to store (and reason about) not only the preferences associated to each domain concept, but also their reliability.

3 A fuzzy approach to store the user profile in an ontology

In a RS the domain ontology permits to classify the objects of recommendation. We consider that each object is an instance of one (or several) of the lowest level classes of the ontology (*i.e.* the leaves). Thanks to the taxonomical structure of the concepts in the ontology, we can reason about the objects at different levels of generality. We propose to use the domain ontology to represent the preferences of the users of the recommender system. In this way, the concepts are interpreted as subsets of the domain in which the user can be interested. As the interest degree can be different from one concept to another, the preferences are represented using fuzzy sets.

Proposition 1. Let us consider a fuzzy set for each concept *c* of the ontology, so that, for each user *u*, $\mu_c(u)$gives the membership degree of *u* to the concept *c*.

This membership degree is personal for each user and represents his degree of interest in a certain concept *c*. If the user is completely interested in *c*, then $\mu_c(u)$=1. Oppositely, when $\mu_c(u)$=0, we assume user *u* is not interested at all in concept *c*.

When a certain user *u* needs a recommendation, we propose to find the values of $\mu_c(u)$for all the concepts in the ontology. Once the ontology has been completely labelled with $\mu_c(u)$, the RS will be able to find the most appropriate items for this user, taking into account that each object is an instance of some of the concepts. The values of $\mu_c(u)$will be calculated using explicit and implicit information elicited from the user interaction with the system. Due to this process of estimation, there is a strong uncertainty in the preference values. To manage this uncertainty, we will consider the following confidence degree:

Proposition 2. Let us consider a confidence level $CL_c(u)$ between 0 and 1 that quantifies the confidence associated to the estimation of the membership degree of *u* to the concept *c*, denoted as $\mu_c(u)$.

A large value of $CL_c(u)$indicates that we can trust the value of $\mu_c(u)$as the true degree of interest of the user *u* for the concept *c*, whereas a low value indicates that the estimation is not so reliable. In that way, not only the degree of membership to the concepts in the ontology is considered to select the best alternatives, but also the confidence on the estimation of those values is taken into account. For instance, the recommender system may decide to ignore the values with a low confidence level, because they have not achieved enough support.

In summary, the user personal profile consists on a copy of the ontology that stores the degree of interest of this user on each concept, as well as the related confidence levels. As an example, let us consider a recommender system for the members of a Hiking association. Fig. 1 shows a small portion of the domain ontology, which can be used to recommend events, news or conferences of interest to the association members. As

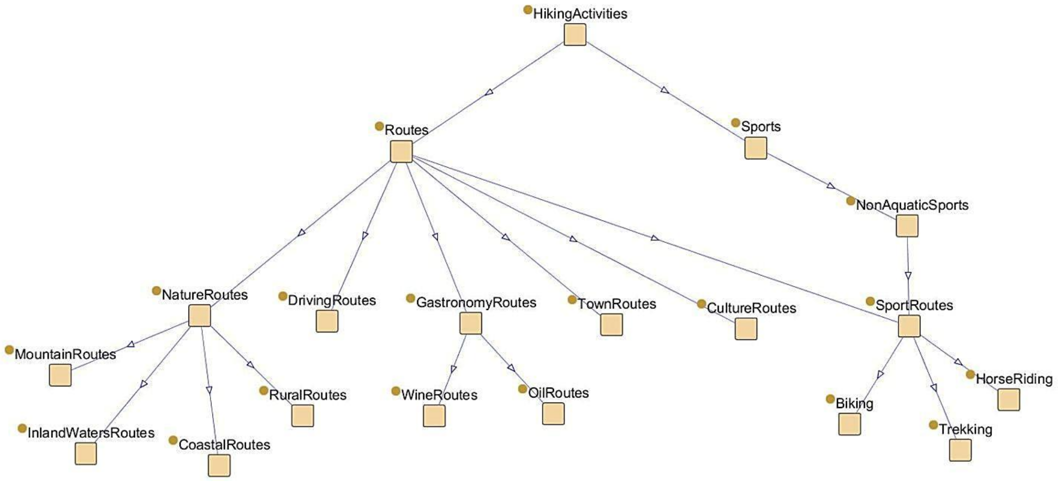


Figure 1.Portion of Hiking ontology.

said before, it is assumed that all the recommendable items are instances of the lowest level concepts (RuralRoutes, WineRoutes, CultureRoutes, Trekking, etc.). The instances do not belong to the profile; they are stored in a database.

3.1 Initialization of the profile

Each ontology concept has an interest degree $\mu_c(u)$estimated by the system, which is calculated from the collection of user information through the session, which can be extracted explicitly or implicitly. For the initialization of the user interests the application asks him to fill in a form wherehe can express the interest on some general domain aspects, represented by first-level ontology concepts (in the example shown in Fig. 1, those general concepts are Routes and Sports). Rating values range from 0.0 (no interest) to 1.0 (highest interest). The confidence level associated to these ratings is 1.0 because the value is fully reliable since it is given directly by the user.

3.2 Propagation of the initial preference and certainty values

The structure of the ontology may be exploited to transfer the preference information through the nodes through a *downwards propagation* of the initial preference and confidence values obtained for the first-level ontology concepts.Imagine that a user explicitly expresses a high interest in the first-level concept Routes($\mu_{Routes}(u)=0.8$,$CL_{Routes}(u)=1.0$). This suggests an interest in different kinds of routes, which are represented by its descendants. Therefore, the system has to transfer the interest shown in the most general concept to its subclasses until the concepts in the lowest level (that are used to instantiate the items of recommendation) are reached. However, there is some level of uncertainty that the interest is equal in all its children, which increases as we propagate to deeper levels of the ontology. We propose to copy the membership degree of the user to the parent class to all its descendants, but decreasing the degree of confidence at each level by a factorα, which can be customized to the needs of the application and represents the decrease in certainty as we move down the ontology hierarchy, far from the general concepts that have been explicitly valued by the user.

Definition 1 (Downwards propagation of the initial preferences)

The preference associated to a concept *c* is calculated as an average of the preferences of his parents (χ^c), weighted by their confidence values. The confidence value associated to *c* is the average of the confidences in his parents, decremented by the factor α:

$$\mu_c(u) = \frac{\sum_{i\in\chi c}\mu_i(u)CL_i(u)}{\sum_{i\in\chi c}CL_i(u)} \qquad (1)$$

$$CL_c(u) = \frac{\sum_{i\in\chi c}CL_i(u)}{|\chi c|} - \alpha \qquad (2)$$

The parameter alpha determines the rate of confidence decrease from one node to its descendent nodes, having α in [0..0,5]. This parameter must be fixed accordingly to each application domain and depending on the maximum number of levels we want to propagate the values. CL is set to 0 if Eq (2) gives a non-negative value.

Table 1.User actions collected by the system.

User actions	Explicit	Implicit	s	w
Save recommended item		●	0.5	0.5
Remove recommended item		●	-0.5	0.5
Request detailed information about an item		●	0.1	0.2
Request item similar to the current one		●	0.15	0.3
Rate an item	●		[-1.0, 1.0]	1.0

4 Dynamic refinement of the user profile

During the execution of the recommender system, we can gather additional knowledge about the user's interests. The evidences provided by the different types of actions on the objects are used to modify both the membership degrees of the user to the related concepts and their confidence level. The information obtained about an object *i* affects the concepts which *i* is instantiating (which are leaves in the ontology).

We distinguish two main types of information that can be obtained from the interaction of the user with the recommender system:

A) Since each object is labelled with concepts at the lowest level of the ontology, we can learn about the interest of the user on these concepts by studying the actions he does on them, which can be either *positive* (*e.g.* saving a recommended item) or *negative* (*e.g.* removing a saved item). For this type of indirect feedback, the confidence level should be low.

B) Recommender systems may ask the user to rate some items shown to him. In this case, the rating values on the items can also be used to estimate the membership degree of the user to the lowest level concepts. The confidence level can be high because this is explicit information provided by the user.

Table 1 summarizes the scores *s* (between -1 and 1) and the weights *w* (between 0 and 1) associated to each user action. This feedback is useful to refine the estimation of the membership degree of the user by inferring his interests based on the behaviour of the user in front of the previously recommended objects. The scores are fixed depending on each application domain, indicating the reward and penalty given to each action. Similarly, the weight of those scores is set on each application case and it is used to control the impact of the actions on the preference scores stored in the user profile.

Assume that we have observed a set of actions A_c on a group of objects that are instances of the concept *c*. The scores and weights associated to these actions are aggregated together as follows:

$$\Delta_c = \frac{\sum_{a\in A_c} s_a w_a}{\sum_{a\in A_c} w_a} \qquad CA_c = \frac{MIN(\lambda, \sum_{a\in A_c} w_a)}{\lambda} \qquad (3)$$

As can be seen in equation 3, the aggregated confidence of the actions is normalized using a parameter λ, which indicates the level above which a higher amount of evidence is not required to have a full aggregated confidence of 1. For instance, it could have a value of 2.4 if 3 good reviews (0.8*3) are considered enough to have a full confidence. If the aggregated confidence in the actions is higher than the current confidence level of the concept ($CA_c \geq CL_c$), then its preference and confidence values are updated as follows:

$$\mu_c = \begin{cases} if\,(\Delta_c > 0) & MIN(1, \mu_c + \beta \times \Delta_c) \\ else & MAX\,(0, \mu_c + \beta \times \Delta_c) \end{cases} \qquad (4)$$

$$CL_c = \beta \times CA_c + (1-\beta) \times CL_c \qquad (5)$$

β is a parameter between 0 and 1 that graduates the level of change between the current values and the scores and weights given by the user actions. The higher its value, the bigger is the impact of the user actions on the concept information. For instance, if it is 0.75, the new confidence will be computed taking into account the confidence in the last action 3 times more (0.25) than the previous confidence.

4.1 Upwards propagation

At this point, the feedback of the user has been used to modify the information stored at the lowest-level concepts of the ontology. After the system has collected a sufficiently large set of user actions, the values can be propagated through the ontology to update the values of other related concepts. In a first step, we make an *upwards propagation* to the ancestor concepts of the modified leaves. Again, the more distant an ancestor is, the more uncertainty we have.

Note that several children of the same concept may have been modified (e.g., the user may have interacted with instances of WineRoutes and OilRoutes, both children of GastronomyRoutes). Let us assume that$\varphi^c$is the set of concepts that are children of *c* and have confidence values higher than a certain threshold (concepts that do not have enough confidence should not influence on their parents). The aggregated preference and confidence values of the children of *c* may be computed as follows:

$$\Delta_c = \frac{\sum_{i\in\varphi^c}\mu_i CL_i}{\sum_{i\in\varphi^c} CL_i} \qquad CA_c = \frac{\sum_{i\in\varphi^c} CL_i}{|\,\varphi^c\,|} \tag{6}$$

If the aggregated confidence of the children of c, CA_c is higher than a threshold, then its preference and confidence values are updated as shown in equations 7 and 8. β is the parameter used in equations 4 and 5, which regulates the degree of change. Those parameters must be empirically studied for each domain.

$$\mu_c = \frac{(1-\beta)\times\mu_c CL_c + \beta\times\Delta_c CA_c}{(1-\beta)\times CL_c + \beta\times CA_c} \tag{7}$$

$$CL_c = \beta\times CA_c + (1-\beta)\times CL_c \tag{8}$$

4.2 Downwards propagation

Once the upwards propagation has been completed, a second step propagates the preference and confidence values to the descendants of the updated nodes. For instance, if the preference of the user in SportRoutes has been modified due to the rating of some HorseRiding activities, a modification of the values for Biking and Trekking seems reasonable, due to their high semantic similarity with HorseRiding.

In this downwards propagation, the information of a concept c is modified according to the preference and confidence values of its parents, χ^c, as long as these confidence values exceed a given threshold. The aggregation of the information of the parents is done equivalently to the upwards case, as follows:

$$\Delta_c = \frac{\sum_{i\in\chi c}\mu_i CL_i}{\sum_{i\in\chi c} CL_i} \quad CA_c = \frac{\sum_{i\in\chi c} CL_i}{|\chi c|} \tag{9}$$

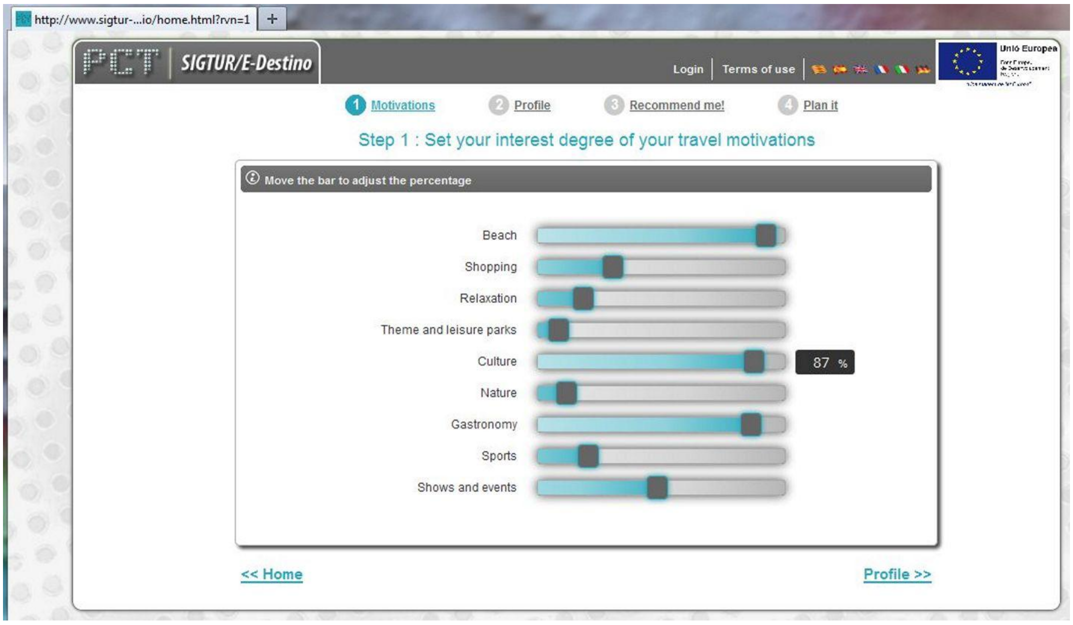
If CA_c is higher than a given threshold and c has not been updated during the upwards propagation, its information is changed according to equations 7 and 8.

5 Recommendation of touristic activities

The ontology-based preference management framework presented in this paper has been implemented in a Web recommender system of tourist activities within the Catalan region of “Costa Daurada and Terres de l’Ebre”, called SIG/Tur. The architecture of the system and its main features are summarized in [2]. The system considers the uncertainty associated to the transmission of general preferences down the ontology hierarchy and the uncertainty associated to the interpretation of the actions of the user.

A specific Tourism domain ontology, focused on the kind of activities available in this particular geographical area, has been manually built, following general guidelines of the World Tourism Organisation. It covers a wide variety of types of activities, which have been classified into nine main concepts that constitute the first level of the hierarchy (see Fig.2). There are 203 concepts in a 5-levels hierarchy.

1,300 activities have been catalogued in an external database, including a textual description, timetable, town and location coordinates, among others. Each activity is annotated with the lowest level concepts in the ontology to which it belongs. The initialization of the user profile is done with the information collected from the tourist with the application form shown in figure 2. In the initialization stage, these values are propagated downwards as explained in section 3.2.

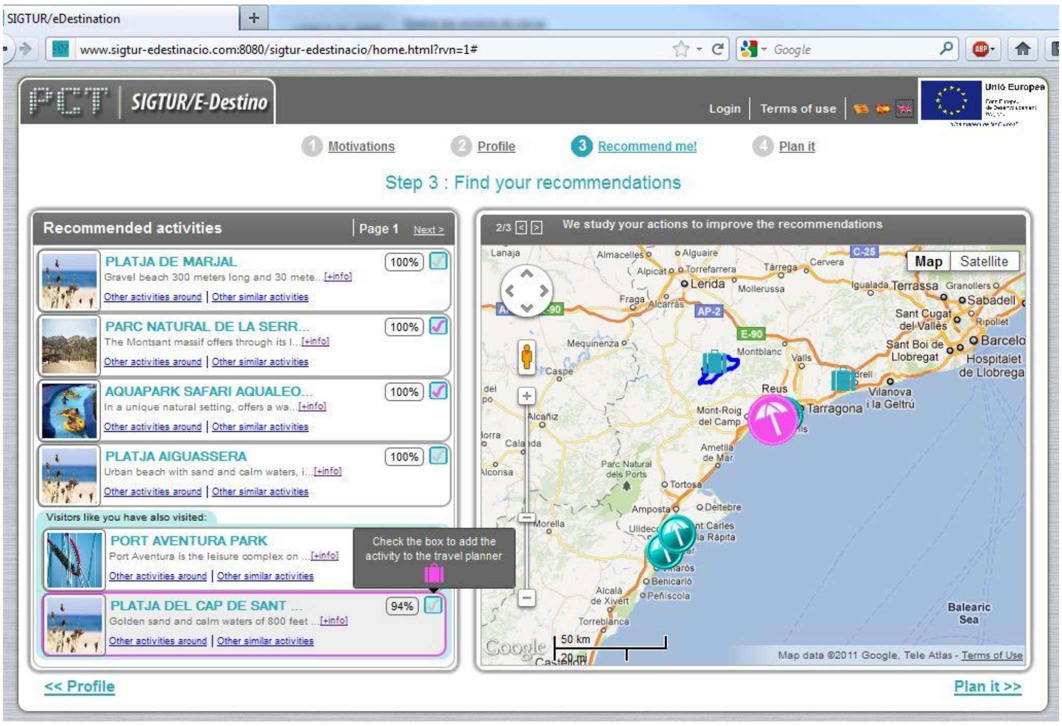


Step 1 : Set your interest degree of your travel motivations

Move the bar to adjust the percentage

Beach
Shopping
Relaxation
Theme and leisure parks
Culture 87 %
Nature
Gastronomy
Sports
Shows and events

Figure 2. Explicit user interests about generic kinds of tourist activities



SIGTUR/eDestination

www.sigtur-edestinacio.com:8080/sigtur-edestinacio/home.html?rvn=1#

SIGTUR/E-Destino

Login | Terms of use

1 Motivations
2 Profile
3 Recommend me!
4 Plan it

Step 3 : Find your recommendations

Recommended activities | Page 1 Next >

PLATJA DE MARJAL 100%
Gravel beach 300 meters long and 30 mete...[+info]
Other activities around | Other similar activities

PARC NATURAL DE LA SERR... 100% ☑
The Montsant massif offers through its l...[+info]
Other activities around | Other similar activities

AQUAPARK SAFARI AQUALEO... 100% ☑
In a unique natural setting, offers a wa...[+info]
Other activities around | Other similar activities

PLATJA AIGUASSERA 100%
Urban beach with sand and calm waters, i...[+info]
Other activities around | Other similar activities

Visitors like you have also visited:

PORT AVENTURA PARK
Port Aventura is the leisure complex on ...[+info]
Other activities around | Other similar activities

Check the box to add the activity to the travel planner

PLATJA DEL CAP DE SANT ... 94%
Golden sand and calm waters of 800 feet ...[+info]
Other activities around | Other similar activities

2/3 We study your actions to improve the recommendations

<< Profile

Plan it >>

Figure 3.SIG/Tur graphical interface.

Using this initial information, a first recommendation is done using both content-based and collaborative-based techniques [5]. The basic idea is that the RS considers the ontology leaves that have a certainty level above a certain threshold, and orders them according to their preference level. After that, the system builds a list of specific activities associated to the concepts of the top of this ranked list and shows them to the user. The system displays the activities and their localization on a map, as shown in Figure 3.The interaction of the user with the recommended items allows refining the user profile dynamically, as described in section 4. Users can select activities which are added to a travel planner, can ask for additional information and also rate the activity proposed. The refinement of the profile is applied after 10 user actions. Then, a new list of recommended activities is proposed to the user.

6 Conclusions

The idea of using semantic domain knowledge to improve the accuracy of the recommendations provided by an intelligent system is compelling, and there are some works that have already suggested the use of ontologies to represent the user profile and the items of recommendation[1, 3, 7, 8]. However, those tools do not support the uncertainty associated to these preferences (both the one due to the lack of initial information and the one associated to the dynamic changes in the user profile induced from the user actions). This work suggests a first step in this direction, considering the maintenance of both preference and certainty information for each ontology concept. The framework is general enough to be usable in different applications, because the system actions (and their scores and weights) and the parameters for preference adaptation can be customized. Our future work includes a thorough analysis of the influence of these parameters in the dynamic change of the user preferences, the analysis of different aggregation procedures of the preferences coming from a set of children/parents, the study of different ways in which the information about preferences and certainties may be used by the RS, the estimation of the user satisfaction with the provided recommendations and the test of this general framework in other domains.

As future work we would like to consider the case of obtaining pairwise comparisons of the actions proposed by the recommender system. In this case, the preference update could consider all the concepts related to the pair of actions compared. Pairwise comparison is a typical approach used to learn the preference model in some Multi-criteria Decision Aid methods.

Acknowledgements

SigTur/E-Destination is a project funded by the FEDER European Regional Funds and the Government of the Province of Tarragona. Part of the work has been supported the Spanish Ministry of Science and Innovation (DAMASK project, Data mining algorithms with semantic knowledge, TIN2009-11005) and the Spanish Government (PlanE, Spanish Economy and Employment Stimulation Plan).

References

[1] Blanco-Fernández, Y., López-Nores, M., Pazos-Arias, J.J. and García-Duque, J.: An improvement for semantics-based recommender systems grounded on attaching temporal information to ontologies and user profiles. Eng. Appl. Art. Intell., 24(8) 1385-1397 (2011)

[2] Borràs, J., de la Flor, J., Pérez, Y., Moreno, A., Valls, A., Isern, D., Orellana, A., Russo, A. and Anton-Clavé, S.: SigTur/E-Destination: A System for the Management of Complex Tourist Regions. In Proc. of International Conference on Information and Communication Technologies in Tourism, ENTER 2011. pp. 39-50. Springer Verlag, Innsbruck, Austria (2011)

[3] Cantador, I. and Castells, P.: Extracting multilayered Communities of Interest from semantic user profiles: Application to group modeling and hybrid recommendations. Computers in Human Behavior, 27(4) 1321-1336 (2011)

[4] Liang, H., Xu, Y., Li, Y., Nayak, R. and Wang, L.T.: Personalized recommender systems integrating social tags and item taxonomy. In Proc. of Web Intelligence and Intelligent Agent Technology, WI-IAT 2009. pp. 540–547. IEEE Press, Milano, Italy (2009)

[5] Montaner, M., López, B. and de la Rosa, J.L.: A Taxonomy of Recommender Agents on the Internet. Artif. Intell. Rev., 19(3) 285-330 (2003)

[6] Nocera, A. and Ursino, D.: An approach to providing a user of a “social folksonomy” with recommendations of similar users and potentially interesting resources. Know.-Based Syst., 24(8) 1277-1296 (2011)

[7] Sieg, A., Mobasher, B. and Burke, R.: Learning Ontology-Based User Profiles: A Semantic Approach to Personalized Web Search. IEEE Intelligent Informatics Bulletin, 8(1) 7-18 (2007)

[8] Sieg, A., Mobasher, B. and Burke, R.: Improving the effectiveness of collaborative recommendation with ontology-based user profiles. In Proc. of Information Heterogeneity and Fusion in Recommender Systems, HetRec 2010. pp. 39-46. (2010)

[9] Tang, X. and Zeng, Q.: Keyword clustering for user interest profiling refinement within paper recommender systems. J. Syst. Soft., 85(1) 87-101 (2012)

[10] Borràs, J., Valls,A., Moreno,A. and Isern,D. Ontology-based management of uncertain preferences in user profiles. in: 14th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2012, Communications in Computer and Information Science, Vol.298, Advances in Computational Intelligence, Part 2, 127-136, Springer Verlag (2012)

Session 7

Invited speaker : Yann Chevaleyre

- LIPN, Université Paris 13
  "***Learning GAI networks***",

  Generalized Additive Independence (GAI) models have been widely used to represent utility functions. In this talk, we will address the problem of learning GAI networks from pairwise preferences. First, we will consider the case where the structure of the GAI network is known of bounded from above. We will see how this problem can be reduced to a kernel learning problem. Then, we will investigate the structure learning problem. After presenting the computational of algorithms can be used to solve this problem.

Session 8

- "*On measuring and testing the ordinal correlation between valued outranking relations*",
  R. Bisdorff[1],
  [1] University of Luxembourg

- "*Elicitation of decision parameters for thermal comfort on the trains*",
  L. Mammeri[1,2], D. Bouyssou[1], C. Galais[2], M. Ozturk[1], S. Segretain[2] and C. Talotte[2]
  [1] CNRS-Lamsade, Université Paris-Dauphine,
  [2] SNCF

- "*Dynamic managing and learning of user preferences in a content-based recommender system*",
  L. Marín[1], A. Moreno[1], D. Isern[1] and A. Valls[1]
  [1] Universitat Rovira i Virgili, Tarragona

- "*An algorithm for active learning of lexicographic preferences*",
  F. Delecroix[1], M. Morge[1], J.-Chr. Routier[1]
  [1] Université Lille 1

On measuring and testing the ordinal correlation between valued outranking relations

Raymond Bisdorff
University of Luxembourg, FSTC/CSC/ILIAS
raymond.bisdorff@uni.lu

Abstract. We generalize Kendall's rank correlation measure τ to valued relations. Motivation for this work comes from the need to measure the level of approximation that is required when replacing a given valued outranking relation with a convenient crisp ordering recommendation.

Keywords: Multiple criteria decision aid; Ordinal correlation; Kendall's tau, Outranking relations; Bipolar credibility valuation.

Introduction

When proposing a measure for providing information on the potentially conflicting nature of the criteria in a given MCDA problem [1], we applied Kendall's rank correlation measure τ to the ordinal comparison of the marginal rankings observed on each criterion. Now, we propose to furthermore generalize the same idea to the direct comparison of bipolarly-valued binary relations [5, 6].

This work is motivated, first, by the need to fine-tune meta-heuristics for multiple criteria based clustering, where the eventual clustering results may be compared to an a priori given pairwise global outranking relation [2]. A second, similar motivation comes from the need to compare multiple criteria based rankings obtained with different ranking rules like Kemeny's, Kohler's, the PROMETHEE net flows rule [14], or, more recently, Dias-Lamboray's prudent leximin rule [3]. Assessing the operational performance of these rules may be based on the more or less consistent ordinal correlation observed between each ranking results and the empirical underlying valued outranking relation.

The present work is closely related to, without being inspired from, recent results concerning the formal and empirical analysis of the fuzzy gamma rank correlation coefficient [4].

After the formal introduction of our correlation measure, and the discussion of some of its properties, we provide empirical results for statistically testing the presence or absence of any correlation between different types of random relations, and more particularly, valued outrankings.

1 Measuring ordinal correlations

1.1 Ordinal correlation between crisp relations

Let R_1 and R_2 be two binary relations defined on the same finite set X of dimension n. Kendall's rank, or ordinal, correlation measure τ is essentially based on the idea of counting the number of concordant (equivalent) non reflexive pairwise relational situations, normalized by the total number $n(n-1)$ of possible such relational situations. If $C = \#\{\, (x,y) \in X^2 \,:\, x \neq y \text{ and } \big((x\,\mathrm{R}_1\,y) \Leftrightarrow (x\,\mathrm{R}_2\,y)\big)\,\}$ denotes the number of concordant non reflexive relational situations we observe, that Kendall's τ measure can be defined as follows[1]:

$$\tau(\mathrm{R}_1, \mathrm{R}_2) \quad := \quad 2 \times \frac{C}{n(n-1)} \; - \; 1\,. \qquad (1)$$

It is worthwhile noticing that Kendall [7, 8] used a very natural way (see [5, 6]) of transforming a direct counting of concordant, i.e. logically equivalent situations, into a bipolarly valued correlation index. Unanimously (100% equivalent situations) concordant relations are matched to a correlation index of value +1.0, 50% concordance between the relations (50% equivalent and 50% not equivalent situations) is matched to a zero-valued correlation index, and unanimously discordant relations

[1] Originally, Kendall [7, 8] counted the number of inversions observed when comparing two linear orders. Formula (1), hence, takes a dual form: $\tau(\mathrm{R}_1, \mathrm{R}_2) = 1 - 2 \times \big(n(n-1) - C\big) \,/\, n(n-1)$ [9, see page 104].

(100% non equivalent situations) are matched to a correlation index of value: -1.0.

Example 1.1. Let us consider the following crisp relations R_1 and R_2 defined on a set $X = \{a, b, c\}$ of nodes, where $R_1 = \{(b, c), (c, a)\}$ and $R_2 = \{(a, b), (a, c), (b, c)\}$. As we observe as many concordant situations: $\neg(b\,\mathrm{R}\,a)$, $(b\,\mathrm{R}\,c)$, and $\neg(c, b)$, as discordant situations: $\neg(a\,\mathrm{R}\,b)$, $(b\,\mathrm{R}\,a)$, and $\neg(c\,\mathrm{R}\,a)$, the Kendall $\tau(\mathrm{R}_1, \mathrm{R}_2)$ correlation index equals: $\tau(\mathrm{R}_1, \mathrm{R}_2) = 2 \times \frac{3}{6} - 1 = 0.0$.

The τ rank correlation index implicitly relies on the assumption that each relation is completely determined. Either $(x\,\mathrm{R}\,y)$ or $\neg(x\,\mathrm{R}\,y)$; all relational situations between any pair of elements of X are exactly known. But, what happens when we compare now valued relations, where the validation of relational situations might be more or less precarious?

1.2 Valued equivalence of relational situations

Let R_1 and R_2 be two binary relations defined on the same finite set X of dimension n and characterized via a bipolar characteristic function r taking values in the rational interval $[-1.0; 1.0]$ [5, 6]. We call such relations, for short, *r-valued* and *of order n*.

For any such valued relation R, its characteristic function r supports the following semantics:

i) $r(x\,\mathrm{R}\,y) = \pm 1.0$ signifies that the relational situation $x\,\mathrm{R}\,y$ is certainly valid $(+1.0)$, resp. invalid (-1.0);

ii) $r(x\,\mathrm{R}\,y) > 0.0$ signifies that the relational situation $x\,\mathrm{R}\,y$ is more valid than invalid;

iii) $r(x\,\mathrm{R}\,y) < 0.0$ signifies that the relational situation $x\,\mathrm{R}\,y$ is more invalid than valid;

iv) $r(x\,\mathrm{R}\,y) = 0.0$ signifies that the relational situation $x\,\mathrm{R}\,y$ is indeterminate, i.e. neither valid, nor invalid.

Logical negation, conjunction, and disjunction of such r-characteristic values may be respectively computed with changing the sign, applying a min, or max operator [5, 6, 10]. For instance:

$$\begin{aligned}
r\big(\neg(x\,\mathrm{R}\,y)\big) &= -r(x\,\mathrm{R}\,y),\\
r\big((x\,\mathrm{R}_1\,y) \wedge (x\,\mathrm{R}_2\,y)\big) &= \min\big(r(x\,\mathrm{R}_1\,y), r(x\,\mathrm{R}_2\,y)\big),\\
r\big((x\,\mathrm{R}_1\,y) \vee (x\,\mathrm{R}_2\,y)\big) &= \max\big(r(x\,\mathrm{R}_1\,y), r(x\,\mathrm{R}_2\,y)\big).
\end{aligned}$$

These logical operators, now, allow us to compute, for instance, the r-valued logical equivalence of any two relational situations:

$$\begin{aligned}
&r\big(\ (x\,\mathrm{R}_1\,y) \Leftrightarrow (x\,\mathrm{R}_2\,y)\ \big)\\
&\quad = r\Big[\big(\neg(x\,\mathrm{R}_1\,y) \vee (x\,\mathrm{R}_2\,y)\big) \wedge \big(\neg(x\,\mathrm{R}_2\,y) \vee (x\,\mathrm{R}_1\,y)\big)\Big]\\
&\quad = \min\Big[\max\big(-r(x\,\mathrm{R}_1\,y), r(x\,\mathrm{R}_2\,y)\big),\\
&\qquad\qquad \max\big(r(-x\,\mathrm{R}_2\,y), r(x\,\mathrm{R}_1\,y)\big)\Big]
\end{aligned}$$

Finally, we will need to measure the average level of determinateness of an r-valued relation R of order n, denoted $d(\mathrm{R})$, and taking value in the interval $[0; 1]$:

$$d(\mathrm{R}) := \frac{\sum_{x,y \in X^2}^{x \neq y} \mathrm{abs}(r(x\,\mathrm{R}\,y))}{n(n-1)}. \qquad (2)$$

Thus, a *crisp* – a completely ± 1-valued – relation shows a determinateness degree of 1, whereas an *indeterminate* – a completely 0-valued – relation shows a determinateness degree of 0.

Example 1.2. We may apply the concepts and tools of this r-valued credibility calculus for assessing, for instance, the actual equivalence of the relational situations we observed in Example 1.1. Take for instance the situation $(a\,\mathrm{R}\,b)$. Here we have: $r(a\,\mathrm{R}_1\,b) = -1.0$ and $r(a\,\mathrm{R}_2\,b) = 1.0$. It follows that $r(a\,\mathrm{R}_1\,b \Leftrightarrow a\,\mathrm{R}_2\,b) = \min(-1.0, 1.0) = -1.0$. Whereas, if we take the pair (b, c), we obtain $r(b\,\mathrm{R}_1\,c \Leftrightarrow b\,\mathrm{R}_2\,c) = \min(1.0, 1.0) = 1.0$. Hence, we faithfully recover in the crisp case, the original Kendall τ values. Suppose now that relation R_1 is not certainly determined and $r(a\,\mathrm{R}_1\,b) = -\alpha$ with $\alpha \in [0; 1]$. In this case $r(a\,\mathrm{R}_1\,b \Leftrightarrow a\,\mathrm{R}_2\,b) = \min(-\alpha, 1.0) = -\alpha$. Similarly, suppose now that $r(b\,\mathrm{R}_1\,c) = \alpha$. In that case $r(b\,\mathrm{R}_1\,c \Leftrightarrow b\,\mathrm{R}_2\,c) = \min(\alpha, 1.0) = \alpha$.

This gives us a hint that the r-valued equivalence of two valued relational situations verifies the following important property:

Property 1.1.
Let R_1 and R_2 be any two r-valued relations defined on the same set X. For all x, y in X^2, we have:

$$\begin{aligned}
&r\big(\,(x\,\mathrm{R}_1\,y) \Leftrightarrow (x\,\mathrm{R}_2\,y)\,\big) =\\
&\quad \pm\min\big(\,\mathrm{abs}(r(x\,\mathrm{R}_1\,y)), \mathrm{abs}(r(x\,\mathrm{R}_2\,y))\,\big).
\end{aligned}$$

Proof.
Suppose $r(x\,\mathrm{R}_1\,y) = \alpha$ and $r(x\,\mathrm{R}_2\,y) = \beta$ with $\alpha, \beta \in [-1; 1]$. If $\mathrm{abs}(r(x\,\mathrm{R}_1\,y)) = \mathrm{abs}(r(x\,\mathrm{R}_2\,y))$, Property 1.1 follows immediately from Equation (2). Otherwise, we may observe the following cases:

1. $|\alpha| > |\beta|$:

   i) if $\alpha > \beta \geq 0$ then $\min[\max(-\alpha, \beta), \max(-\beta, \alpha)] = \beta > 0$;

   ii) if $\alpha > 0 > \beta$ then $\min[\max(-\alpha, \beta), \max(-\beta, \alpha)] = \beta < 0$;

   iii) if $\beta > 0 > \alpha$ then $\min[\max(-\alpha, \beta), \max(-\beta, \alpha)] = -\beta < 0$;

   iv) if $0 \leq \beta > \alpha$ then $\min[\max(-\alpha, \beta), \max(-\beta, \alpha)] = -\beta > 0$.

2. $|\beta| > |\alpha|$:

   i) if $\beta > \alpha \geq 0$ then $\min[\max(-\alpha, \beta), \max(-\beta, \alpha)] = \alpha > 0$;

   ii) if $\alpha > 0 > \beta$ then $\min[\max(-\alpha, \beta), \max(-\beta, \alpha)] = -\alpha < 0$;

iii) if $\beta > 0 > \alpha$ then
$\min[\max(-\alpha,\beta),\max(-\beta,\alpha)] = \alpha < 0$;

iv) if $0 \leq \alpha > \beta$ then
$\min[\max(-\alpha,\beta),\max(-\beta,\alpha)] = -\alpha > 0$.

□

With Property 1.1 in mind, we may now generalize Kendall's ordinal correlation measure for taking into account genuine r-valued relations.

1.3 Correlations between valued relations

The r-valued equivalence of relational situations may be judiciously used, as in the crisp case, for assessing the numerator of the ordinal correlation measure. Yet, stating the adequate denominator needs some further going considerations. In the classical crisp case, following Kendall, we divide the sum of pairwise equivalences with $n(n-1)$, i.e. the total number of concerned non reflexive situations. If we would proceed this way in the valued case, the resulting measure would integrate a mixture of both the ordinal correlation as well as the actual determinateness of the equivalence observed between the considered r-valued relations. To factor out both these effects we take, instead, as denominator the maximum possible sum of r-valued equivalences we could potentially observe when both r-valued relations would show completely concordant relational situations.

Hence, we formulate the r-valued ordinal correlation measure τ between two r-valued relations R_1 and R_2, defined on a same set X, as follows: $\tau(\mathrm{R}_1,\mathrm{R}_2)$:=

$$\frac{\sum_{x\neq y} r\Big(\,(x\,\mathrm{R}_1\,y) \Leftrightarrow (x\,\mathrm{R}_2\,y)\,\Big)}{\sum_{x\neq y} \min\Big[\,\mathrm{abs}\,\big(r(x\,\mathrm{R}_1\,y)\big), \mathrm{abs}\,\big(r(x\,\mathrm{R}_2\,y)\big)\Big]} \tag{3}$$

where, in order to avoid divisions by zero, we assume that a zero sum of r-valued equivalences occurring in the numerator always takes strong precedence over the potential zero sum determinateness occurring in the denominator. Indeed, if the sum of absolute values of r-valued equivalences is zero, then so must essentially be the sum of the corresponding signed r-valued equivalences.

It is furthermore worthwhile noticing that the denominator in Formula (3), once divided by the number of non reflexive relational situations, i.e.:

$$\frac{\sum_{x\neq y} \min\Big[\,\mathrm{abs}\,\big(r(x\,\mathrm{R}_1\,y)\big),\, \mathrm{abs}\,\big(r(x\,\mathrm{R}_2\,y)\big)\Big]}{n(n-1)} \tag{4}$$

gives, in fact, the average determinateness degree of the r-valued equivalence relation $\mathrm{R}_1 \Leftrightarrow \mathrm{R}_2$ observed between both r-valued relations. In case of crisp relations, this determinateness degree always takes maximum value 1.0. But, as soon as one of both valued relations appears completely indeterminate, $d(\mathrm{R}_1 \Leftrightarrow \mathrm{R}_2)$ becomes 0. In this latter case, $\tau(\mathrm{R}_1,\mathrm{R}_2)$ becomes equally 0. Otherwise, $\tau(\mathrm{R}_1,\mathrm{R}_2)$ gives the ordinal correlation measure independently of their equivalence determinateness level.

Describing the ordinal correlation between two r-valued binary relations, hence, requires to show both, the relative ordinal correlation measure τ defined in Equation (3), as well as the determinateness degree D of the corresponding relational equivalence defined in Equation (2).

Example 1.3. To illustrate this insight, we consider in Table 1, two randomly r-valued relations R_1 and R_2 of order $n = 3$ and defined on a same set $X = \{a,b,c\}$. The pairwise r-valued equivalence situations $\mathrm{R}_1 \Leftrightarrow \mathrm{R}_2$ are shown in Table 2.

Table 1. Examples of randomly valued relations

$r(x\,\mathrm{R}_1\,y)$	a	b	c
a	$-$	$+0.68$	$+0.35$
b	-0.94	$-$	$+0.80$
c	-1.00	$+0.36$	$-$

$r(x\,\mathrm{R}_2\,y)$	a	b	c
a	$-$	-0.32	$+0.58$
b	-0.14	$-$	$+0.75$
c	-1.00	$+0.08$	$-$

Table 2. r-valued equivalence between R_1 and R_2

$r(x\,\mathrm{R}_1\,y \Leftrightarrow x\,\mathrm{R}_2\,y)$	a	b	c
a	$-$	-0.32	$+0.35$
b	$+0.14$	$-$	$+0.75$
c	$+1.00$	$+0.08$	$-$

Hence,

$$\begin{aligned}
&\tau(\mathrm{R}_1,\mathrm{R}_2)\\
&\quad= \frac{-0.32+0.35+0.14+0.75+1.00+0.08}{+0.32+0.35+0.14+0.75+1.00+0.08}\\
&\quad= \frac{0.200}{0.264} \;=\; +0.7575\,,
\end{aligned}$$

whereas the corresponding equivalence determinateness:

$$d(\mathrm{R}_1 \Leftrightarrow \mathrm{R}_2) \;=\; \frac{0.264}{6} \;=\; 0.44\,.$$

Thus, nearly 76% or the jointly determined ordinal information is actually shared by both r-valued relations, independently of the respective 44% of determinateness of the r-valued equivalence situations.

If we had instead used the classical denominator $n(n-1)$ for computing the actual correlation measure, we would have obtained a much smaller τ value of only: $\frac{0.200}{6} = +1/3$ (33.33% instead of 75.75%); potentially misleading us, thus, on the apparent correlation between R_1 and R_2. Notice that this result of 1/3 is in fact the product of $\tau(\mathrm{R}_1, \mathrm{R}_2)$ with $d(\mathrm{R}_1, \mathrm{R}_2)$, i.e. 0.7575×0.44.

1.4 Properties of the ordinal correlation measure

Again, let R_1 and R_2 be two r-valued binary relations defined on a set X of dimension n. We say that R_1 and R_2 show *a same*, respectively *an opposite*, orientation if, for all non reflexive pairs (x, y) in X, $r(x \, \mathrm{R}_1 \, y \Leftrightarrow x \, \mathrm{R}_2 \, y) > 0$, respectively $r(x \, \mathrm{R}_1 \, y \Leftrightarrow x \, \mathrm{R}_2 \, y) < 0$.

Property 1.2. *If two r-valued relations R_1 and R_2, defined on the same set X, show a same, respectively an opposite, orientation, $\tau(\mathrm{R}_1, \mathrm{R}_2)$ equals $+1.0$, respectively -1.0, independently of their equivalence determinateness $d(\mathrm{R}_1, \mathrm{R}_2)$.*

Proof. Property 1.2 readily follows from Property 1.1 and the observation that in case of a same orientation, respectively an opposite orientation, the sum of terms in the numerator of Formula (3) equals the sum, respectively the negation, of the sum of terms in the denominator. □

The *logical negation* of an r-valued relation R, denoted $\neg\,\mathrm{R}$, is called its *dual* relation. And, the *reciprocal* of an r-valued relation R, denoted Я, is called its *converse* relation. The following very natural properties are verified by the generalized ordinal correlation measure τ.

Property 1.3. *Let R_1 and R_2 be two r-valued binary relations defined on a same set X and let the ordinal correlation measure τ be defined by Formula (3):*

$$
\begin{aligned}
\tau(\mathrm{R}_1, \mathrm{R}_2) &= \tau(\mathrm{R}_2, \mathrm{R}_1) && (5)\\
\tau(\neg\,\mathrm{R}_1, \mathrm{R}_2) &= -\tau(\mathrm{R}_1, \mathrm{R}_2) && (6)\\
\tau(\text{Я}_1, \text{Я}_2) &= \tau(\mathrm{R}_1, \mathrm{R}_2) && (7)\\
\tau(\neg\,\text{Я}_1, \neg\,\text{Я}_2) &= \tau(\mathrm{R}_1, \mathrm{R}_2) && (8)
\end{aligned}
$$

Proof. Equations (5) to (8) follow immediately from the definition of the τ correlation measure (see Formula 3):

(5) Symmetry of the τ measure follows from the commutativity of the max and min operators used for computing the terms of numerator as well as denominator.

(6) Negating one of the r-valued relations changes solely the sign of all r-valued equivalences in the numerator.

(7) Taking the converse relations of both r-valued relations means correspondingly transposing all (x, y) terms to (y, x) terms, jointly in numerator and denominator; thus, leaving invariant the resulting fraction.

(8) Taking the codual relations, i.e. the negation of the converse, of both r-valued relations, hence, leaves invariant their τ correlation measure.

□

In order to avoid, the case given, being fooled by randomness, we address in the next section the problem of estimating via Monte Carlo simulations the actual significance of the ordinal correlation measure when working with different types of random r-valued relations.

2 Testing for ordinal correlations

Originally, Kendall only considered correlations between crisp rankings without ties. Kendall's τ measure for pairs of random instances of such rankings of order n is known to show an expected correlation $\mu_\tau = 0.0$ with standard deviation [12]:

$$\sigma_\tau = \sqrt{\frac{2(2n+5)}{9n(n-1)}}.$$

This gives for rankings of order $n = 20$, for instance, a standard deviation $\sigma_\tau \approx 0.17$. Assuming a nearly Gaussian distribution of μ_τ, we obtain 90% and 99% confidence intervals of approximatively ± 0.22, respectively ± 0.40. Hence, a measure $|\tau| > 0.4$ observed between two rankings of 20 objects reveals a significant positive or negative ordinal correlation between them.

To similarly estimate the significance of the τ correlation measure when comparing r-valued relations, we were running extensive Monte Carlo simulations with, in turn, three specific models of random relations, namely random uniformly r-valued ones, r-valued weak tournaments and, randomly generated bipolar outranking relations resulting from the aggregation of multiple cost and benefit criteria.[10].

2.1 Random r-valued relations

First we consider a model of random relations where to each non reflexive pair of elements (x, y) in X is associated a uniform random float between -1.0 and 1.0. Each possible r-valued relation has thus the same probability to appear. To get a hindsight on the correlation and determinateness measures we may obtain with this genuine kind of r-valued relations, we generate large samples of 100 000 pairs of such r-valued relations for different orders n. Each pair (x, y) has, thus, in the limit, an average probability of 1/2 to be related or not; the strict indeterminate value 0.0 having no chance to effectively appear as random number.

In Figure 1 is represented the scatter plot of the resulting tuples (d, τ) for r-valued relations of order 20. What strikes immediately is the nearly perfect symmetry of the resulting distributions, both of the determinateness degrees, as well as of the correlation measures.

In this model of random r–valued relations, the distribution of the equivalence measures E of each pairwise relational situation is following a symmetric triangular density with spread ± 1.0 and

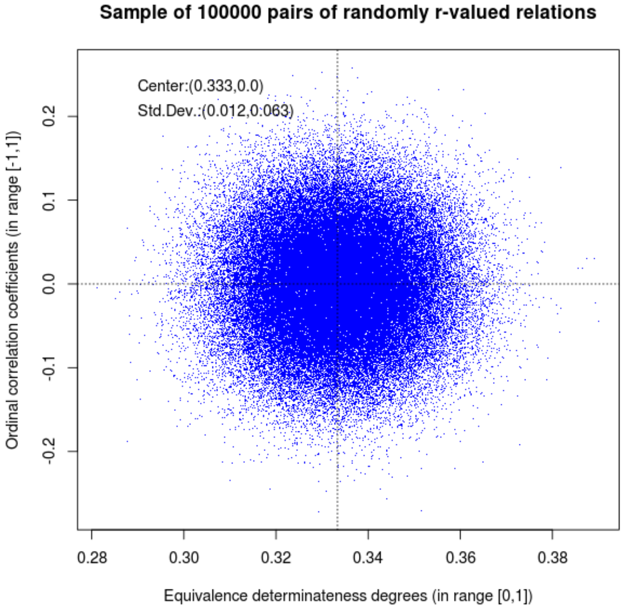


Figure 1. Scatter plot of (d,τ) for pairs of randomly r-valued relations of order 20

0 mode. Such random variables admit a mean $\mu_e = 0$ and a standard deviation $\sigma_e = \sqrt{3/18}$. A similar situation is observed when considering their equivalence determinateness measures. Each term in the denominator of Formula 3 is chosen from a same independent and identically distributed random variable D with positive density $1-x$ for x in $[0;1[$. This distribution – a special case of the triangular distribution where the mode equals the lower limit – shows a mean $\mu_d = 1/3$ and a standard deviation of $\sigma_d = \sqrt{1/18}$.

Hence, the observed random ordinal correlation measures $\tau = \frac{\sum E}{\sum D}$ result from the ratio of two non-independent sums of $n(n-1)$ independent and identically distributed random variables. Following from the central limit theorem, the observed statistics (see Table 3) rapidly show, with increasing order n, a more and more Gaussian distribution with mean $\hat{\mu}_\tau \approx \frac{\mu_e}{\mu_d} = 0$ and standard deviation:

$$\frac{\hat{\sigma}_\tau}{\sqrt{n(n-1)}} \approx \frac{\sigma_e}{\mu_d}\frac{1}{\sqrt{n(n-1)}}$$

getting ever smaller with increasing order n of the r-valued relations.

In Table 3 we may notice that the observed empirical standard deviations $\hat{\sigma}_d$ when multiplied with $\sqrt{n(n-1)}$ converge indeed to σ_d which equals $\sqrt{1/18} = 0.2357023$. Similarly, we may notice that the observed standard deviation $\hat{\sigma}_\tau$ tends also to the theoretical standard deviation $\sigma_\tau = 3\sqrt{3/18} = 1.224745$ when multiplied by $\sqrt{n(n-1)}$. Notice, however, a consistent negative bias of roughly 2%.

Example 2.1. Consider two given r-valued relations R_1 and R_2 of order 20. To test if they could have been randomly generated, we may apply a two-sided test with null hypothesis: H_0 "relations R_1 and R_2 are randomly r-valued". From the empirical results, we see that H_0 may be rejected with an error probability of 10% when $|\tau(\mathrm{R}_1,\mathrm{R}_2)| > 0.1035$ or $|d(\mathrm{R}_1,\mathrm{R}_2) - 0.3333| > 0.202$. Using, our theoretical standard deviations $\sigma_\tau = 3\sqrt{3/18}$, respectively $\sigma_d = \sqrt{1/18}$, we may precisely confirm these confidence intervals: ±0.1033, respectively ±0.0199 with a Gaussian test.

We have thus established a generic test apparatus for two-sided, or both positive or negative one-sided tests for measuring the significance of the ordinal correlation and equivalence determinateness of any two given r-valued relations.

Yet we are more interested in testing the correlation and equivalence determinateness when considering a specific subset of r-valued relations, namely weakly complete ones. Uniformly r-valued random relations, indeed, are statistically quite regularly structured, with on average 1/4 of double links, 1/4 of single forward links, 1/4 of single backward links and, 1/4 of no links. When working in the fields of social choice or multiple criteria decision aid with "*at least as good as*" preferential situations, we usually consider complementary concordance versus discordance relations [11] that do not allow a "*no link*" situation.

2.2 Random weakly complete relations

Formally we say that a r-valued relations R is weakly complete if for all $(x,y) \in X$, $r(x\,\mathrm{R}\,y) < 0$

Table 3. Summary Statistics, for 100000 pairs of randomly r-valued relations

$d(\mathrm{R}_1, \mathrm{R}_2)$	$\overline{d}$	$\hat{\sigma}_d$	$\hat{\sigma}_d\sqrt{n(n-1)}$	Conf. 90%	Conf. 99%
$n = 5$	0.3333	0.0527	0.23568	±0.0866	±0.1355
$n = 10$	0.3334	0.0249	0.23622	±0.0406	±0.0645
$n = 15$	0.3333	0.0162	0.23476	±0.0266	±0.0418
$n = 20$	0.3333	0.0121	0.23587	±0.0202	±0.0276
$n = 30$	0.3333	0.0080	0.23597	±0.0132	±0.0207
$n = 50$	0.3333	0.0048	0.23758	±0.0078	±0.0121
$\tau(\mathrm{R}_1, \mathrm{R}_2)$	$\overline{\tau}$	$\hat{\sigma}_\tau$	$\hat{\sigma}_\tau\sqrt{n(n-1)}$	Conf. 90%	Conf. 99%
$n = 5$	0.0003	0.2731	1.22134	±0.4500	±0.6766
$n = 10$	0.0000	0.1289	1.22285	±0.2181	±0.3291
$n = 15$	0.0000	0.0842	1.22017	±0.1386	±0.2156
$n = 20$	0.0000	0.0621	1.21055	±0.1035	±0.1425
$n = 30$	0.0000	0.0414	1.22113	±0.0681	±0.1064
$n = 50$	0.0000	0.0247	1.22259	±0.0406	±0.0636

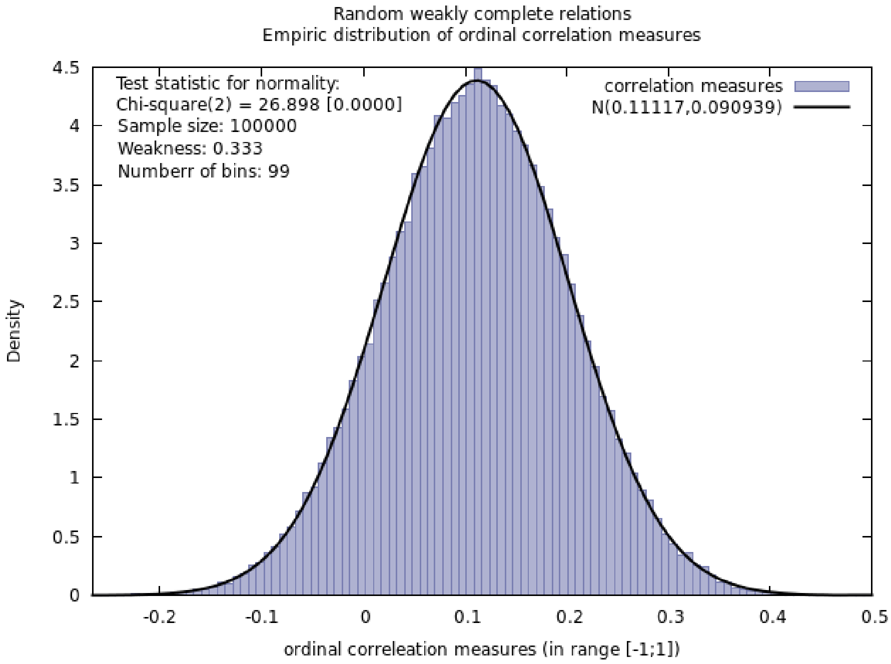


Figure 2. Histogram of correlation measures with normality test for pairs of random weakly complete relations of order 20

implies $r(x\,\mathrm{R}\,y) \geq 0$. Each link is therefore either a double, or a single forward or backward link, each one with equal probability 1/3.

Determinateness distribution of equivalence of pairs of this model of random r-valued relations remains very close to 1/3, as in the general model above (see Table 4), except a slight lowering of its mean values (compare with Table 3). Similarly, we may again observe an empiric distribution of correlation measures which follows, with increasing order of the relations, more and more, due to the central limit theorem, a Gaussian distribution. In Figure 2 is represented a histogram from a sample of 100 000 random instances of weakly complete r-valued relations of order 20. Notice first the fact that the sampled mean correlation measure $\hat{\mu}_\tau$ is shifted roughly by +0.111, depending on the given degree of weakness. In the limit, a weakness degree of 1.0, on the one hand, would give always the same complete relation, showing, hence, a constant correlation measure of 1.0. A weakness degree of 0.0, on the other hand, would give samples of random tournaments with mean and median correlation measures concentrated around 0 as in the general case above.

In Table 4 we summarize empiric statistical results for weakly complete r-valued relations of different orders, maintaining constant a weakness degree of 1/3. The observed distribution of correlation measures τ, besides the already mentioned positive shift of the mean by approximately +0.111, also shows an empiric standard deviation $\hat{\sigma}_\tau$ multiplied by $\sqrt{n(n-1)}$ that is no longer a constant independent of the given order n. Hence,

Table 4. Summary Statistics, for 100000 pairs of random weakly (1/3) complete relations

$d(\mathrm{R}_1, \mathrm{R}_2)$	$\hat{\mu}_d$	$\hat{\sigma}_d$	$\hat{\sigma}_d\sqrt{n(n-1)}$	Conf. 90%		Conf. 99%	
$n = 5$	0.33344	0.05268	0.23568	0.24920	0.42208	0.20493	0.47489
$n = 10$	0.33316	0.02490	0.23622	0.29262	0.37433	0.27069	0.39861
$n = 15$	0.33316	0.01634	0.23476	0.30646	0.36025	0.29164	0.37578
$n = 20$	0.33320	0.01209	0.23587	0.31342	0.35315	0.30262	0.36486
$n = 30$	0.33316	0.00799	0.23597	0.32006	0.34630	0.31269	0.35406
$n = 50$	0.33318	0.00477	0.23758	0.32537	0.34102	0.32099	0.34558
$\tau(\mathrm{R}_1, \mathrm{R}_2)$	$\hat{\mu}_\tau$	$\hat{\sigma}_\tau$	$\hat{\sigma}_\tau\sqrt{n(n-1)}$	Conf. 90%		Conf. 99%	
$n = 5$	0.1112	0.3032	1.3560	−0.3981	+0.6039	−0.6559	+0.8229
$n = 10$	0.1113	0.1592	1.5103	−0.1537	+0.3713	−0.3019	+0.5082
$n = 15$	0.1112	0.1138	1.6491	−0.0767	+0.2978	−0.1810	+0.3978
$n = 20$	0.1112	0.0909	1.7720	−0.0395	+0.2604	−0.1231	+0.3399
$n = 30$	0.1116	0.0681	2.0087	−0.0003	+0.2234	−0.0631	+0.2869
$n = 50$	0.1112	0.0484	2.3957	+0.0313	+0.1905	−0.0132	+0.2348

the equivalence measures on the numerator of the τ measures are no longer independent and identically distributed. Consequently, the central limit theorem is no longer automatically applicable. When verifying the plausibility of the randomness hypothesis when comparing weakly complete r-valued relations, we are thus solely left with the potentially biased sample standard deviations and the corresponding tail percentiles estimations.

Example 2.2. With 99 bins, the χ^2 test (see Figure 2), however, clearly confirms ($26.898 \ll \chi^2(0.01, 98) = 68.396$, p-value $= 0.0$) for order $n = 20$ and average weakness 1/3, the quality of the Gaussian approximation with empirical mean $\hat{\mu}_\tau = 0.1112$ and standard deviation $\hat{\sigma}_\tau = 0.0909$. The Gaussian 90%-confidence, resp. 99%-confidence, interval of the mean correlation measure μ_τ, hence, gives the limits $[-0.0384; +0.2606]$, respectively $[-0.1231; +0.3454]$. And, both intervals are, indeed, very close to the empirical ones (see Table 4, row $n = 20$) we obtain with a sample of 100 000 random instances.

Finally, we consider a special subset of weakly complete r-valued relations, namely r-valued *outranking* relations.

2.3 Random outranking relations

Concordance relations, i.e. weakly complete r-valued relations naturally result from the ordinal aggregation of multiple performance criteria when considering the weighted concordance of the statements: "*x performs at least as good as y*" [10]. Our random model for such kind of r-valued relations is based on randomly generated performances for all decision actions in $x \in X$ on each criterion. We distinguish three types of decision actions: *cheap*, *neutral* and *expensive* ones with an equal proportion of 1/3. We also distinguish two types of weighted criteria: *cost* criteria to be *minimized*, and *benefit* criteria to be *maximized*; in the proportions 1/3 respectively 2/3. Random performances on each type of criteria are drawn, either from an ordinal scale $[0; 10]$, or from a cardinal scale $[0.0; 100.0]$, following a parametric triangular law of mode: 30% performance for cheap, 50% for neutral, and 70% performance for expensive decision actions, with constant probability repartition 0.5 on each side of the respective mode. Cost criteria use mostly cardinal scales (3/4), whereas benefit criteria use mostly ordinal scales (2/3). The sum of weights of the cost criteria always equals the sum weights of the benefit criteria. On cardinal criteria, both of cost or of benefit type, we observe following constant preference discrimination quantiles: 5% indifferent situations, 90% strict preference situations 90%, and 5% veto situation. We call this random model of r-valued relations for short random *CB-outranking* relations.

In Table 5 we summarize the empirical results for various numbers of decision actions (n) and criteria (c). Most noticeable is here the diminishing average determination degrees with rising numbers n of actions and, especially numbers c of criteria. Indeed, the fixed proportion of veto situations (5%) on each cardinal criteria augments, with the number of criteria, the probability of the presence of pairwise indeterminate, i.e. 0-valued, outranking situations. Furthermore, the empiric distribution of the determination degrees appears no more to converge to a Gaussian type limit.

In Figure 3, one may notice, indeed, on a sample of 100 000 random CB-outranking relations of order $n = 20$ and criteria $c = 13$, an apparent left asymmetry, confirmed by a positive skewness of 0.876, as well as a clearly leptocurtic distribution (excess kurtosis: +1.7884) of the observed determinateness degrees. Comparing the observed distribution with a theoretical gamma distribution, reveals a positive match with parameters: $\alpha = 38.119$ and $\beta = 0.004$. With order $n = 30$ and criteria $c = 21$ one obtains a similar gamma estimation with parameters: $\alpha = 64.594$ and $\beta = 0.002$.

With rising numbers of indeterminate preferential situations, the proportion of double links compared to single links, is no more as regular (1/3 against 2/3) as in the genuine model of random weakly complete relations. In the $n = 20$ and

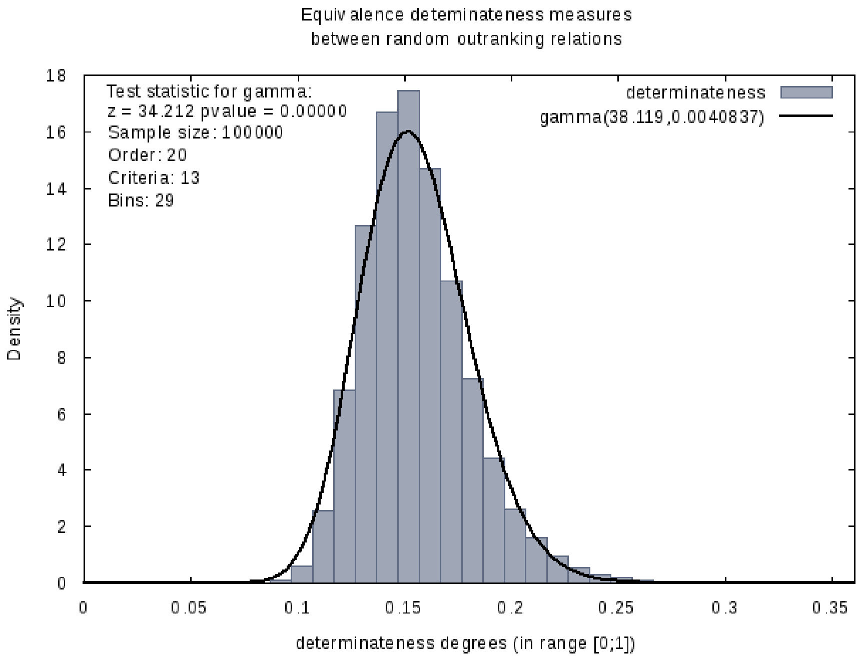


Figure 3. Histogram of determination degrees for pairs of random CB-outranking relations of order 20

Table 5. Summary Statistics, for 100000 pairs of random CB-outranking relations

$d(\mathrm{R}_1, \mathrm{R}_2)$	$\hat{\mu}_d$	$\hat{d}_{50\%}$	$\hat{\sigma}_d$	Conf. 90%		Conf. 99%	
$n = 5, c = 3$	0.3259	0.3250	0.1131	0.1500	0.5255	0.0750	0.6333
$n = 10, c = 7$	0.2207	0.2165	0.0482	0.1494	0.3072	0.1204	0.3681
$n = 15, c = 9$	0.1910	0.1867	0.0362	0.1399	0.2577	0.1196	0.3102
$n = 20, c = 13$	0.1557	0.1527	0.0252	0.1203	0.2013	0.1053	0.2435
$n = 30, c = 21$	0.1372	0.1357	0.0174	0.1120	0.1674	0.1002	0.1989
$\tau(\mathrm{R}_1, \mathrm{R}_2)$	$\hat{\mu}_\tau$	$\hat{\tau}_{50\%}$	$\hat{\sigma}_\tau$	Conf. 90%		Conf. 99%	
$n = 5, c = 3$	0.0378	0.0345	0.5145	-0.7929	$+0.8610$	-1.0000	$+1.0000$
$n = 10, c = 7$	0.0629	0.0644	0.3037	-0.4420	$+0.5560$	-0.6483	$+0.7467$
$n = 15, c = 9$	0.0727	0.0761	0.2417	-0.3323	$+0.4667$	-0.5206	$+0.6354$
$n = 20, c = 13$	0.0984	0.1017	0.2085	-0.2492	$+0.4383$	-0.4224	$+0.5904$
$n = 30, c = 21$	0.1239	0.1272	0.1712	-0.1639	$+0.4007$	-0.3162	$+0.5339$

$c = 13$ case, with a sample of 100 000 random instances, we observe only 18.7% double links, with 71.6% single links and, in this particular case, 9.8% of indeterminate links. Consequently, in Table 5, we observe lower average correlation measures $\hat{\mu}_\tau$ than with the previous model. Furthermore, the convergence to a Gaussian limit distribution with rising order of the relations is no more apparent when considering in Figure 4 the Q-Q plot of the simulated correlation quantiles against Gaussian quantiles for the case $n = 20$ and $c = 13$. A very high χ^2 value (567.356) rejects, indeed, the Gaussian approximation hypothesis.

Appreciating the significance of the correlation between pairs of CB-outranking relations remains, hence, solely possible on the basis of sampled tail percentiles from Monte Carlo simulations. In the appendix we have gathered estimated 5%, 95%, 0.5% and 99.5% percentiles for relations of various orders and numbers of criteria that may be relevant in an MCDA context.

Example 2.3. Let us eventullly consider the random r-valued CB-outranking relation shown in Table 6. Relation R_1, with an average determination degree $d(R_1) = 0.397$, is defined on $n = 10$ decision actions and results from the ordinal concordance observed on $c = 7$ performance criteria. Applying for instance Kemeny's ranking rule [13] would give us the following crisp linear order: $[4, 2, 7, 8, 9, 1, 10, 6, 3, 5]$, showing a highly significant correlation of $+0.888$ with R_1 (see the upper limit 0.747 of the 99% confidence interval in Table 5). Indeed, under the hypothesis of a completely random ordering, such a high correlation measure would appear in less than 0.5% of cases. When ranking now with the help of the net flows scores à la PROMETHEE [14], we would obtain the order: $[4, 9, 2, 7, 8, 10, 1, 6, 3, 5]$, showing a less higher correlation $(+0.776)$ with R_1.

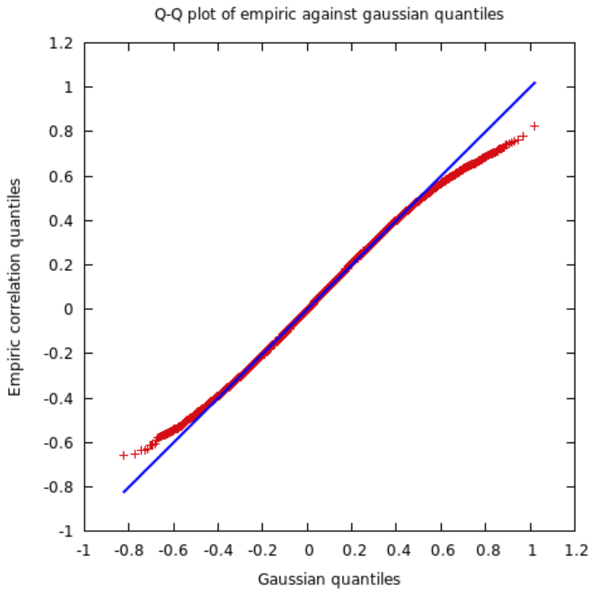


Figure 4. Q-Q plot of empiric again normal correlation quantiles for pairs of random CB-outranking relations of order 20

Table 6. Example of random CB-outranking relation ($n = 10$, $c = 7$)

R_1	1	2	3	4	5	6	7	8	9	10
1	$-$	$+0.14$	$+0.43$	-0.14	$+0.29$	$+0.14$	$+0.43$	-0.14	± 0.00	$+0.43$
2	$+0.43$	$-$	$+0.43$	-0.43	$+0.43$	$+0.14$	$+0.14$	$+0.43$	$+0.14$	$+0.14$
3	-0.43	-0.43	$-$	-0.71	$+0.43$	$+0.00$	-0.43	-1.00	-1.00	-0.14
4	$\mathbf{+0.14}$	$\mathbf{+0.71}$	$\mathbf{+1.00}$	$-$	$\mathbf{+0.71}$	$\mathbf{+0.43}$	$\mathbf{+0.71}$	$\mathbf{+0.43}$	$\mathbf{+0.14}$	$\mathbf{+0.57}$
5	$+0.14$	-0.43	-0.43	-0.71	$-$	-0.71	-1.00	$+0.14$	-1.00	-0.43
6	-0.14	-0.14	$+1.00$	-0.43	$+0.71$	$-$	-0.14	-0.14	$+0.14$	-0.43
7	$+0.14$	$+0.14$	$+0.43$	-0.43	$+1.00$	$+0.14$	$-$	$+0.14$	$+0.43$	$+0.29$
8	$+0.43$	-0.14	$+1.00$	-0.43	$+0.43$	$+0.14$	-0.14	$-$	$+0.43$	-0.14
9	$+1.00$	-0.14	$+1.00$	-0.14	$+1.00$	$+0.43$	$+0.14$	-0.14	$-$	$+0.14$
10	-0.43	-0.14	$+0.43$	$+0.14$	$+0.43$	$+0.43$	$+0.29$	$+0.14$	-0.14	$-$

Kohler's rule would, furthermore, give us the order: $[4, 2, 8, 10, 9, 6, 1, 7, 3, 5]$ with the same correlation of $+0.776$. Finally, Tideman's ranked pairs rule [15], in fact the dual of Dias and Lamboray's leximin rule [3], will deliver the order: $[4, 2, 8, 9, 1, 7, 10, 6, 3, 5]$ with, this time again, a highly significant correlation measure of $+0.872$. As we compare here each time R_1 with a 1.0-valued (crisp) linear order, the correlation underlying equivalence determination degree is actually, in all cases, equal to $d(R_1) = 0.397$. By the way, we may notice that the same first ranked decision action with all the ranking rules is action 4, in fact a CONDORCET winner that outranks all other decision actions with a majority of at least 57% of the criteria weights (see row 4 in Table 6).

Conclusion

We have consistently generalized Kendall's rank correlation measure τ to r-valued binary relations via a corresponding r-valued logical equivalence measure. The so extended ordinal correlation measure, besides remaining identical to Kendall's measure in the case of completely determined linear orders, shows interesting properties like its independence with the actual determinateness degree of the r-valued equivalence. Empirical confidence intervals for different models of random r-valued relations, like weakly complete and, more particularly, r-valued outranking relations are elaborated.

References

[1] Bisdorff R (2008) On clustering the criteria in an outranking based decision aid approach. In Le Thi H A et al. (eds) Modelling, Computation and Optimization in Information Systems and Management Sciences Springer-Verlag CCIS 14:409–418 1

[2] Bisdorff R and Meyer P and Olteanu A (2011) A Clustering Approach using Weighted Similarity Majority Margins. In Tang J et al (Eds) Advanced Data Mining and Applications ADMA 2011 Part I Springer-Verlag LNAI 7120:15-28 1

[3] Dias L C and Lamboray C (2010) Extensions of the prudence principle to exploit a valued outranking relation. European Journal of Operational Research 201(3):828–837 1, 9

[4] Ruiz D and Hüllermeier E (2012) A formal and empirical analysis of the gamma rank correlation coefficient. Information Sciences 206:1–17 1

[5] Bisdorff R (2000) Logical foundation of fuzzy preferential systems with application to the electre decision aid methods. Computers and Operations Research 27:673-687 1, 2

[6] Bisdorff R (2002) Logical Foundation of Multicriteria Preference Aggregation. In: Bouyssou D et al (eds) Essay in Aiding Decisions with Multiple Criteria. Kluwer Academic Publishers 379–403 1, 2

[7] Kendall M G (1938) A New Measure of Rank Correlation. Biometrica 30:81–93 1

[8] Kendall M G (1955) Rank Correlation Methods. Hafner Publishing Co New York 1

[9] Degenne A (1972) Techniques ordinales en analyse des données statistique. Hachette Paris 1

[10] Bisdorff R (2012) On polarizing outranking relations with large performance differences. Journal of Multi-Criteria Decision Analysis DOI: 10.1002/mcda.1472 1–20 2, 4, 7

[11] Bouyssou D and Pirlot M (2005) A characterization of concordance relations. European Journal of Operational Research, 167(2):427–443 5

[12] Valz P D and McLeod A I (1990) A Simplified Derivation of the Variance of Kendall's Rank Correlation Coefficient. The American Statistician 44(1) 39–40 4

[13] Lamboray C (2007) A comparison between the prudent order and the ranking obtained with Borda's, Copeland's, Slater's and Kemeny's rules. Mathematical Social Sciences 54:1–16 8

[14] Brans JP and Vincke Ph (1985) A preference ranking organisation method : The PROMETHEE method for MCDM. Management Science 31(6):647–656 1, 8

[15] Lamboray C (2009) A prudent characterization of the Ranked Pairs Rule. Social Choice and Welfare 32:129–155 9

Elicitation of decision parameters for thermal comfort on the trains

Mammeri Lounes[+*], **Bouyssou Denis**[*], **Galais Cedric**[+], **Ozturk Meltem**[*], **Segretain Sandrine** [+], **Talotte Corine**[+],

+: SNCF, email: <nom>@sncf.fr

***: Lamsade/CNRS, Universite Paris Dauphine, email: <nom>@lamsade.dauphine.fr,**

Abstract. We present in this paper a real world application for the elicitation of decision parameters used in the evaluation of thermal comfort in high speed trains. The model representing the thermal comfort is a hierarchical one and we propose to use different aggregation methods for different levels of the model. The methods used are rule-based aggregation, Electre Tri and 2-additive Choquet. We show in this paper the reasons of the choice of such methods and detail the approach used for the elicitation of the parameters of these methods.

1 Introduction

Comfort is one of the main raison of the choice of trains for long trips. In this paper we are interested in one of the composant of global comfort which is the thermal one. We show how we define the thermal comfort using physical evaluations (temperature, air speed, etc.) in order to be as close as possible to the comfort perception of train passengers. In the following section we present how we establish our model. Our model requires different aggregation steps, in Section 3 we introduce these aggregation steps by presenting in a brief way their formulations, the raisons of their choice and specially the approach that we used for the elicitation of their decision parameters. We conclude our paper by some recommendations for elicitation approaches.

2 Thermal comfort model

Existing methods used for the evaluation of the thermal comfort on high speed trains are based on the Fanger's model ([4]), initially developed for office buildings. Fanger's model uses two indices, the *PMV* (Predicted Mean Vote) and the *PPD* (Predicted Percentage of Dissatisfied). The *PMV* is calculated using five criteria : clothing, metabolic rate (activity of the subject), temperature, air velocity and humidity, and is devised on the basis of tests conducted on a large group of subjects. Once the *PMV* value has been established from tables, it is then possible to determine the *PPD*. Fanger's model is devoted to static situations with long time exposure. The climatic environment parameters and activities of subjects are supposed to be constant. For these reasons its use for trains is not always very adequate. Moreover, some recent research done by the SNCF ([22], [16]), specially some surveys with the passengers on the train, showed that the results of the Fanger's model are not always correlated with the perception of the passengers. Figure 1 presents an example of responses of five passengers to the question “How do you evaluate the thermal conditions in this train?” and the evaluation given by the *PMV*. The first part of the figure represents the answers of passengers and the second part the results obtained by the *PMV*.

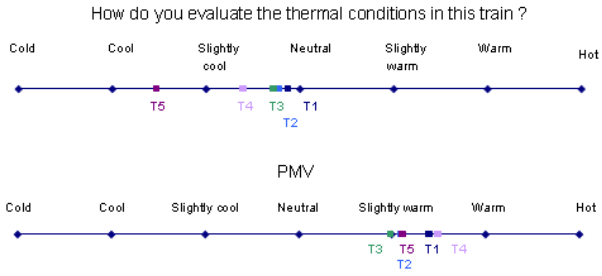


Figure 1. Difference between observations and the *PMV* results

A bibliographical summary of previous projects and research studies on evaluation and perception of thermal comfort was carried at the SNCF [26]. Some of these studies show that there are some perceptive parameters, missed in Fanger's model, which must be taken into account in the evaluation of thermal comfort.

1. The thermal comfort is a subjective notion, the perception can change from a subject to another one and this variability is not taken into account by the *PMV* index. Indeed, although this variability may be estimated by the *PPD*, it is not possible to estimate the thermal comfort of a given subject or a group of subjects sharing the same perception of the comfort.
2. The thermal preferences of a passenger may change with the season.

Other research studies done by the SNCF ([21], [27]) showed that the comfort on the trains is closely related to two perceptions:

1. there must be *no unpleasant sensations* caused by climatic parameters during the journey,
2. there must not be a *discontinuity of ideal thermal conditions*. Such discontinuity is generally caused by the variations of outside temperature, drafts air and the gap between outside and inside temperature.

Another result of these studies is that the most important climatic parameters are temperature and air speed.

Using these results and after several meetings with thermal experts we propose the following model presented in Figure 2 for the evaluation of thermal comfort on high speed trains:

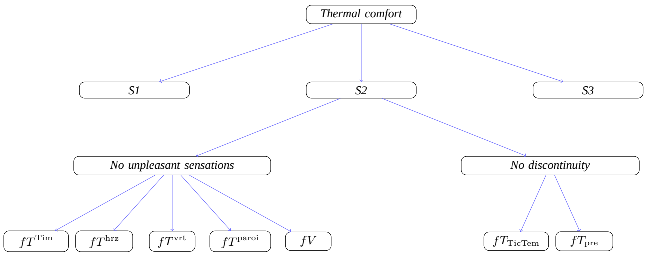


Figure 2. *Thermal comfort model*

In Figure 2,

- S_i represents a season,
- fT^{Tim} represents the weighted mean gap between the average inside temperature t and the reference inside temperature t_{ideal},
- fT^{hrz} represents the weighted mean gap between the maximum and minimum inside temperature on a horizontal section,
- fT^{vrt} represents the weighted mean gap between the maximum and minimum inside temperature measured at head, chest and legs of the passenger,
- fT^{paroi} represents the weighted mean gap between the temperature next to the windows and the average inside temperature,
- fV represents the mean of normalized gaps between the reference range of air speeds and the inside air speeds,
- fT_{TicTem} represents the average rate of change of inside temperatures according to the outside temperatures,
- fT_{pre} represents the gap between the inside temperature and the reference inside temperature when a passenger enters into a train.

3 Agregations

Each node of the thermal comfort model needs an aggregation method, the aggregations are done from the bottom to the top of the model.

3.1 Procedure to find the most appropriate aggregation method

In order to find the most appropriate aggregation method for each node, we ask the following questions to the experts:

Q_1: do we need a ranking or a classification of trains on this node?
Q_2: do we have ordinal or quantitative data?
Q_3: are there any dependance between the subcriteria of this node?
Q_4: is it acceptable to have a compensation between the subcriteria of this node?
Q_5: are there some important subcriteria of this node which may have a veto power (it means that such a subcriteria may put a veto for a good global evaluation if the evaluation of the train is not sufficient for this special subcriteria)?
Q_6: are there too many subcriteria in this node?

Table 3.1 presents a quick analysis of three aggregation methods (rule-based aggregation, Electre tri, Choquet) in relation to the previous questions. These methods will be presented in the following subsections. The answers are given in a very general way, some of them may be different with additional studies (for instance if we have ordinal data, we can translate the ordinal evaluation to utilities with a good elicitation method, ...).

Question	Rule-based	Electre	Choquet
Q_1	classification	classification	ranking
Q_2	ordinal/quantitat.	ordinal/quantitat.	quantitat.
Q_3	dependance	no dependance[1]	dependance
Q_4	no compensat.	no compensat.	compensat.
Q_5	veto[2]	veto	no veto
Q_6	not too many[3]	5-6 criteria[4]	5-6 criteria[5]

Table 1. methods and their properties

The choice of the aggregation is done in accordance with the answers to the questions presented above but there are also two other points that we have to take into account. The method must:

1. provide results in accordance with the preferential expectations of thermal experts and the answers of passengers to the surveys,
2. be easy to understand. It means that if there are many aggregation methods with expected properties, we may chose the most intuitive one in order to facilitate the use and the acceptation of the method by thermal experts. For thermal experts rule-based method is the most intuitive one between the three aggregation methods of Table 3.1 (they are used to have logical rules for the evaluations). However, the logical rules which will be used must be easy to interpret, it is not acceptable to have a big number of rules which have not intuitive meaning but correspond to the answers of the passengers to the survey. After rule-based method the experts fell more comfortable with Electre tri method since all the parameters (weights, indifference thresholds, veto thresholds, etc.) are present in a transparent way while some important indices of Choquet integrals (dependance and importance indices) are not very transparent in the beginning of the evaluations.

These two last points are important for our approach. The first point may be used in the validation step of our approach by comparing the theoretical results with passengers answers. It also says us that we can use some preferential examples in order to determine the parameters of the chosen aggregation method. This point is central for the following section where we will present the elicitation methods. The last point says us that we have to see first of all if we can use rule-based aggregation with simple rules if not we have to try Electre tri and finally we have to test Choquet integrals.

In the following we will present the aggregation method used in each node of the model. For confidentiality purposes we can provide neither the real examples that we used during the elicitation steps nor the real values of decision parameters.

3.2 No discontinuity

In this node we have two criteria $fT^{\mathrm{TicTem}}(e,S)$ and $fT^{\mathrm{pre}}(e,S)$ (see Figure 2), evaluated on cardinal scales, to be minimized. The experts stated that they just need to have a classification into three ordinal categories “no discomfort”, “mild discomfort” and “discomfort”. Our idea is to find a simple aggregation procedure like a small set of rules because of the small number of criteria and categories.

The classical rule-based methods in multicriteria decision making (MCDA) have their roots in rough sets theory [23] which aims at providing a set of rules $R = \{R_1, R_2, .., R_k\}$ (“if <conditions> then

<decision>") from a learning set of decision examples provided by the DM. This learning set is a set of alternatives A evaluated on a set of attributes $Q = \{q_1, .., q_m\}$ for which, decisions (the assignment of alternatives into categories $\{C_1, .., C_p\}$) were taken in the past by the DM (some fictitious examples may be also used if there are no previous decisions). The difficulty of the classical rough sets approach for MCDA is that it can not deal with preference order on the elements of Q and the categories $\{C_1, .., C_p\}$, and thus may violate *the monotonicity* of preferences. For this purpose, Greco, Mattarazzo and Slowinski proposed a generalization of the classical approach by proposing what they called Dominance-based Rough Set Approach (DRSA) [7, 8]. In our application we do not need DRSA since the induction of rules could be performed directly with the DM because of the small number of criteria, categories and also because of some limit values that the experts used to have. However, we should ensure that the set of rules satisfies three properties : the exclusiveness (each alternative must be assigned at most to one category), the monotonicity (the set of rules must be coherent with the dominance principle) and the exhaustivity (each alternative a must be assigned to a category by a rule) [6].

It turned out that the rules inducted with the experts by using a small set of fictitious examples, make use of the *minimum* aggregator. Indeed, for the experts, each criterion has two thresholds (s_1 and s_2 for $fT^{\mathrm{TicTem}}(e, S)$ and s'_1 and s'_2 for $fT^{\mathrm{pre}}(e, S)$) separating three comfort categories (reflecting three levels of comfort like in Tab. 2).

Level of comfort	Ordinal values
No discomfort	3
Mild discomfort	2
Discomfort	1

Table 2. *The coding of comfort categories*

Let $Cl^{\mathrm{TicTem}}(e, S)$ and $Cl^{\mathrm{pre}}(e, S)$ be the translations of $fT^{\mathrm{TicTem}}(e, S)$ and $fT^{\mathrm{pre}}(e, S)$ in terms of levels of comfort :

$$Cl^{\mathrm{TicTem}}(e, S) = \begin{cases} 3 & \text{if } fT^{\mathrm{TicTem}}(e, S) \le s_1 \\ 2 & \text{if } s_1 < fT^{\mathrm{TicTem}}(e, S) \le s_2 \\ 1 & \text{if } fT^{\mathrm{TicTem}}(e, S) > s_2 \end{cases} \quad (1)$$

$$Cl^{\mathrm{pre}}(e, S) = \begin{cases} 3 & \text{if } fT^{\mathrm{pre}}(e, S) \le s'_1 \\ 2 & \text{if } s'_1 < fT^{\mathrm{pre}}(e, S) \le s'_2 \\ 1 & \text{if } fT^{\mathrm{pre}}(e, S) > s'_2 \end{cases} \quad (2)$$

After that a train is assigned to one of three comfort categories for *No discontinuity* using the minimum operator:

$$Cl^{\mathrm{NoDisc}}(e, S) = \min \left\{ Cl^{\mathrm{TicTem}}(e, S), Cl^{\mathrm{pre}}(e, S) \right\}$$

3.3 No unpleasant sensations

In this node we have five criteria (see Figure 2) evaluated on cardinal scales, to be minimized. The experts stated that here again they just need a classification into three ordinal categories "no discomfort", "mild discomfort" and "discomfort" (see again Table 2).

After some discussions with the experts on comfort and on rolling stocks about these criteria, they claim that the first and the last criterion ($fT^{\mathrm{Tim}}(e, S)$ and $fV(e, S)$) are by far the most important and can not be compensated by the three others for reducing the discomfort sensation.

The fact that we need an ordinal classification by using five criteria (it is too much to have intuitive logical rules) and that we have some type of veto (and/or no compensation), are the basic motivations of the choice of Electre Tri method in this node.

Electre Tri is a multicriteria sorting method developed by B. Roy [24]. Its principle is to assign alternatives to predefined and strictly ordered categories (from the worst to the best): $C_1, C_2, ...C_{p+1}$. The assignment of an alternative $a \in A$ in a category is based on the comparison of a with the profiles $b_1, b_2, ..., b_p$ (which separate these categories) on m criteria $g_1, g_2, .., g_m$. A profile b_h is a fictitious alternative which is considered as the lower limit of the category C_{h+1} for $h = 1, .., p$. The comparison of an alternative a with a profile b_h is performed with an outranking relation S, whose meaning is *"a is at least as good as b"*. The assertion aSb is validated if and only if the two following conditions are satisfied: a "majority" of criteria is in favor of a (the weighted sum of criteria in favor of a is greater than a threshold) and none of the criterion which is in favor of b should be against (put a veto) this assertion.

The parameters that can be inferred for a Electre Tri model are:

- The weights of the criteria $k_1, .., k_m$
- The profiles $g_i(b_h)$ $\forall i$ and $\forall h$
- The veto thresholds v_i $\forall i$ (if there are)
- The indifference q_i and preference p_i thresholds $\forall i$ (if there are)

These preferential parameters can be either provided by the DM himself, which rarely happens, or inferred by aggregation/desaggregation methodologies. In these methodologies, the DM is asked to provide a holistic judgment about a subset of potential alternatives $A^* \subset A$ by assigning them in predefined categories. Often, a mathematical programming is solved in order to obtain the estimated parameters that best restore the assignment proposed by the DM, we can have two possible cases for that:

- the mathematical programming can restore the assignment: then the DM can see the results of assignment of other potential alternatives by the inferred model, which can help him to provide further informations or
- the mathematical programming can not restore the assignment: then the DM can see which preferences are inconsistent with the model (but not necessary with its reasoning), so, he may either modify (or withdraw) them or decide that these preferences are so important that the model of Electre Tri must be dropped.

The main difficulty when inferring an Electre Tri model with a mathematical programming is that we can not infer all the parameters simultaneously because the corresponding constraints are nonlinear and non-convex. Therefore some parameters must be inferred directly with the DM.

In the literature, the first methodology for inferring Electre Tri parameters by mathematical programming was performed by V. Mousseau and R. Slowinski [19]. In 2001, V. Mousseau, J. Figueira and J.P. Naux proposed a linear programming formulation for inferring the weights [18]. A. Ngo The and V. Mousseau in 2002 proposed an elicitation of the category limits [28]. Besides the aggrega-

[6] Vanderpooten and Azibi [1] have proposed an approach to check if the rule base satisfies the three previous requirements, provided that the rules have a particular structure. This approach consists on transforming the rules from logical to algebraic representation which allows to solve a series of linear programming in order to check the three requirements. We can also identify with this approach, alternatives which are not covered by rules satisfying these requirements.

tion/desegregation methodologies, direct methods was performed for inferring the Electre Tri model, like SRF ([5]).

We used the following procedure for our application:

- Since we need three categories, we just need to define two profiles. The profiles b_1 and b_2 were determined directly with the experts because they are used to work with some comfort levels defined by the limit evaluations on criteria.
- The weight of criteria were elicited by identifying all subsets of minimum coalitions of criteria $\underline{F} \subseteq F$ in favor of an alternative such that the alternative remains at least as good as the profile for the experts, without taking into account the veto power of criteria (for instance the expert says that it is sufficient to have a better than b for the three first criterion in order to say that a is at least as good as b, ...)
- Ones the weights are determined, we considered a set of learning alternatives in order to elicitate the veto thresholds.

$$A^* = \bigcup_{\underline{F} \subseteq F} (A_{\underline{F}}(1) \cup A_{\underline{F}}(2))$$

built from $\underline{F}$ such that for $p = \{1, 2\}$:

$$\begin{aligned} A_{\underline{F}}(p) = \{a \in A : \forall i \in \underline{F} : g_i(a) > g_i(b_p) \text{ and} \\ \forall j \in F \backslash \underline{F} : g_j(a) \leq g_j(b_1)\} \end{aligned}$$

We focused then on some alternatives $a \in A^*$ for which, we increase progressively the value of $g_j(a)$ (we decrease the performance) $\forall j \in F \backslash \underline{F}$, keeping the same performances in the remaining criteria, until the assertion aSb_p being not valid. Let g_j^* be the smallest value such that aSb_p is not valid. The veto threshold is thus:

$$v_j(g_j(a)) = g_j^* - g_j(a)$$

3.4 Comfort in a given season

In this note the evaluations on the *"No unpleasant sensations"* and the*"No discontinuity"* will be aggregated. These two criteria are evaluated on an ordinal scale with three grades. The experts stated that here again they just need to have a classification into three ordinal categories "no discomfort", "mild discomfort" and "discomfort" (see again Table 2). The small number of criteria and grades allows us to use rule-based methods in a very similar way as in Subsection 3.2. When asking the experts about the importance of the criteria, they stated that the *no unpleasant sensations* criterion is more important. Our second questioning was to know if the overall discomfort in a given season is greater than the maximum discomfort arising from *"No unpleasant sensations"* and *"No discontinuity"*. The answer was negative and besides, they stated that the overall discomfort is close enough to the maximum discomfort arising from the two criteria (the smallest category among the two criteria).

On the basis of this preferential information, we thought it could be useful to keep the same number of categories (the discomfort does not increase) as an evaluation of thermal comfort on each season. The principle of the set of rules is to assign the alternatives to the worst category among the categories corresponding to the two criteria in witch the alternative is assigned excepted when the alternative is in the *category* 3 for *"No unpleasant sensations"* and *category* 1 for *"No discontinuity"*, in which case it is assigned to the *category* 2. This set of rules can be summarized in the following formula:

$$Cl^{\text{season}}(e,S) = \begin{cases} 2 \text{ if } Cl^{\text{NoUnplSns}}(e,S) = 3 \text{ and } Cl^{\text{NoDisc}}(e,S) = 1 \\ \min\{Cl^{\text{NoUnplSns}}(e,S), Cl^{\text{NoDisc}}(e,S)\} \text{ otherwise} \end{cases} \tag{3}$$

3.5 Thermal comfort

The aggregation in this node will provide the global evaluation of thermal comfort. We have to aggregate three evaluations, each of them being ordinal with three grades representing the thermal comfort in season S_i. We began our discussion with experts by trying to define an aggregation which will provide three ordinal classes as in other nodes. We tried first of all rule-based methods. Intuitively, we though that the *minimum* operator would be a good candidate. However, some pairwise comparison examples that we showed to our experts proved that the *minimum* operator was not adequate. Moreover, it was not possible to find simple rules in accordance with their preferences. Then, we tried to see if we could use another classification method such as Electre Tri. The main difficulty of such an approach was the fact that for the experts it was very difficult to define a semantic for the categories. Futhermore, during the discussions with experts we noticed that there may be some dependancies between the three seasons. For that reason we decided to test Choquet integrals by proposing some pairwise comparisons to our experts.

Choquet integral in MCDA is an aggregation operator developed by T. Murofushi and M. Sugeno at the end of the eighteenth [25, 20]. Since, many studies and applications of Choquet integral in MCDA have been carried mainly for building the theoretical foundations [13, 12, 11, 15] and eliciting the parameters [6]. Choquet integral is a generalization of the most known scoring methods: *The weighted sum, the ordered weighted average* [29], *the weighted minimum and the maximum* [3]. Choquet integral of an alternative a, evaluated on the family of criteria F is given by the following formula:

$$C_\mu(a) = \sum_{i=1}^{n} \left[a_{\sigma(i)} - a_{\sigma(i-1)}\right] \mu(\sigma(i), ..., \sigma(n)) \tag{4}$$

Where $a_i = u_i(g_i(a))$.
$u_i(.) : X_i \mapsto [0, 1]$ are non decreasing utility functions.
σ is a permutation on F such that: $a_{\sigma(1)} \leq a_{\sigma(2)} \leq ... \leq a_{\sigma(n)}$.
$\mu(.)$ is a *capacity* on F

Definition 1 *A capacity (or a fuzzy measure) μ on F is a set function*

$$\mu(.) : \mathcal{P}(F) \mapsto [0, 1]$$

satisfying the following conditions:

(i) $\mu(\emptyset) = 0, \mu(F) = 1$
(ii) $\forall S, T \subseteq F : S \subseteq T \Rightarrow \mu(S) \leq \mu(T)$

The capacity $\mu(S)$ of a subset of criteria S can be interpreted as the weight importance of the coalition of criteria of S. It allows to consider more preferential information than the scoring methods mentioned below, like the interactions among criteria or the mutual dependence of criteria.

Choquet integral provides also some numerical indices for analyzing the preferential information like *the Shapley value* $\Phi_\mu(i)$ for measuring the importance of a criterion and the interaction index $I_\mu(S)$ for measuring the interaction among the criteria belonging to $S \subseteq F$.

$$\Phi_\mu(i) = \sum_{T \subseteq F \backslash i} \frac{(n - |T| - 1)!|T|!}{(n)!} [\mu(T \cup i) - \mu(T)] \tag{5}$$

$$I_\mu(S) = \sum_{T \subseteq F \backslash S} \frac{(n - |T| - |S|)!|T|!}{(n - |S| + 1)!} \sum_{L \subseteq S} (-1)^{|S|-|L|} \mu(T \cup L) \tag{6}$$

where n is the cardinality of F. The application of Choquet integral in MCDA requires the elicitation of utility functions u_i and the capacities μ. The main requirement when eliciting the utility functions u_i is the *commensurability* of the scales. The MACBETH approach [2] is often used for eliciting the utility functions (by assuming that the DM is able to give information using intensity of preference) by building an interval scale u_i in order to encode the attractiveness of the elements of subsets $\overline{X}_i = \{(0_1, .., 0_{i-1}, x_i, 0_{i+1}, .., 0_m)/x_i \in X_i\}$ s.t. 0_i and 1_i are the worst and the best values in X_i. The commensurability is ensured by fixing the scales: $u_i(1_i) = 1$ and $u_i(0_i) = 0 \; \forall i$.

Regarding the elicitation of the capacities μ, several methodologies and algorithms have been developed in the literature. The general idea of these methodologies is to ask the DM to express his preferences on a set of learning alternatives A^*. These preferences from which the capacities will be elicited, can be a partial ranking of:

- Alternatives of A^*
- Differences of intensity of preferences of some alternatives in A^*
- Importance of criteria,
- Interactions between criteria
- ...

This preferential information is then translated into mathematical constraints such as:

- $a \succ b \Rightarrow C_\mu(a) \geq C_\mu(b) + \delta_1$
- $a \succ b$ more than $c \succ d \Rightarrow C_\mu(a) - C_\mu(b) \geq C_\mu(c) - C_\mu(d) + \delta_1$
- The criterion i is more important than the criterion $j \Rightarrow \Phi_\mu(i) \geq \Phi_\mu(j) + \delta_2$
- The criteria i and j are complementary $\Rightarrow I_\mu(ij) \geq \delta_3$
- ...

Where δ_1, δ_2 and δ_3 are preference thresholds which must be defined with the DM. It is also possible to fix the number of criteria which may interact.

Definition 2 (k-additivity) *A capacity μ on F is k-additive if there is no interaction among criteria of every subset $S \subseteq F$ whose cardinality is greater than k, ie., $\forall S \subseteq F$ s.t. $|S| > k$, $I_\mu(S) = 0$.*

Most of the methodologies in the literature [14, 10, 9, 17] use an optimization problem with the previous constraints for identifying the capacities. The objective function may differ from a methodology to another.

If a solution is found to this optimization problem then the DM can analyze the results corresponding to the Choquet integral with the identified capacities, he may add further preferential information and thus solves again a new optimization problem. Such a process is performed iteratively until finding a satisfactory model.

If no solution is found to the optimization problem then either the DM preferences are not consistent with the theoretical properties of Choquet integral (transitivity of preferences, monotonicity...etc.) or the number of parameters to be identified is not sufficient to restore the DM preferences. In the first case, inconsistencies must be detected and the DM must change its preferences. In the second case we increase progressively the additivity of the capacities until a solution is found.

The inference of the parameters of the Choquet integral model consists on the elicitation of the utility functions and the capacities:

Elicitation of utility functions The utility functions $u_S(Cl^{\text{season}}(e, S))$ (where $S \in \{S1, S2, S3\}$) corresponding to the criteria of our problem Cl^{season} which will be aggregated with Choquet integral must be commensurate. This requirement leads to put for each criterion $u_S(1) = 0$ (the worst evaluation) and $u_S(3) = 1$ (the best evaluation). We put then $u_S(2) = 0.5$ after ensuring that the difference of attractiveness $u_S(2) - u_S(1)$ is equivalent to $u_S(3) - u_S(2)$ for $S \in \{S1, S2, S3\}$. We thus have:

$$u_S(Cl^{\text{season}}(e, S)) = \begin{cases} 1 \text{ if } Cl^{\text{season}}(e, S) = 3 \\ 0.5 \text{ if } Cl^{\text{season}}(e, S) = 2 \\ 0 \text{ if } Cl^{\text{season}}(e, S) = 1 \end{cases}$$

Let us remark that for our application the elicitation of the utility functions was not problematic because the criteria are evaluated on homogeneous scales and the set X_i of the possible values of the criteria is small. In general cases, this step is more difficult.

Elicitation of the capacities The elicitation of the capacities was performed as follows:

1- **Collecting the preferential information**: We first asked the experts and the DMs to provide an order representing the importance of criteria (which season is important?) in order to build a "relevant" set of learning examples. We built 14 fictitious trains. This set of learning examples was a set of pairwise comparisons of some of these trains. We asked then the experts and the DMs to give their preferences related to this set.

2- **Interactions among criteria**: We have transformed the set of pairwise comparisons into linear constrains and we tried to find additive capacities ($k = 1$) which corresponds to the weighted sum. When solving a linear programming with these constraints, we found no solution. The reason of such failure may be the presence of some types of interactions among criteria. Hence we decided to test a 2-additive model.

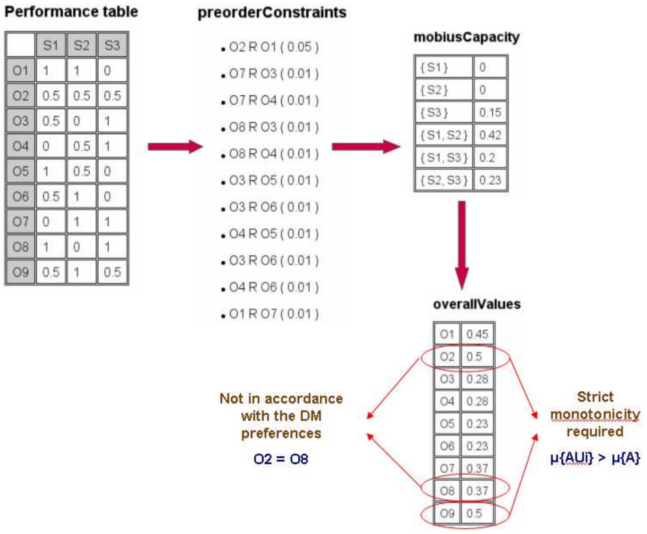
3- **Aggregation/disaggregation procedure**: In order to find the capacities which best restore the preferences, we have used an aggregation/disaggregation procedure. We have first fixed the additivity to 2 (interactions only among pairs of criteria) and we tried to find 2-additive capacities by an approach proposed by Marichal and Roubens [14] which aims at solving a linear programming where the objective function to be maximized is the minimal difference between the Choquet integrals of the compared alternatives. The linear programming have the following form:

$$(LP) \begin{cases} \max f = \epsilon \\ C_\mu(a_1) - C_\mu(b_1) \geq \delta_1 + \epsilon \\ C_\mu(a_2) - C_\mu(b_2) \geq \delta_2 + \epsilon \\ . \\ . \\ . \\ \mu(\emptyset) = 0 \\ \mu(F) = 1 \\ \mu(S) \leq \mu(T), \forall S \subseteq T, \forall T \subseteq F \end{cases} \tag{7}$$

The linear programming gave us several feasible solutions representing the capacity values of the Choquet integral. We chose the first solution and used it in order to obtain a total weakorder of our 14 fictitious trains. We asked then the experts and the DMs if this weakorder was in accordance with their preferences (see Fig. 3). The answer was negative because:

- there were trains which were dominated by others while they had the same overall value (the alternatives O2 and O9 in Fig. 3),

- there were some trains (on which the DMs were not asked to express their preferences) having different overall values while they were considered as equivalent by the experts (the alternatives O2 and O8 in Fig. 3).


Performance table

	S1	S2	S3
O1	1	1	0
O2	0.5	0.5	0.5
O3	0.5	0	1
O4	0	0.5	1
O5	1	0.5	0
O6	0.5	1	0
O7	0	1	1
O8	1	0	1
O9	0.5	1	0.5

preorderConstraints

- O2 R O1 (0.05)
- O7 R O3 (0.01)
- O7 R O4 (0.01)
- O8 R O3 (0.01)
- O8 R O4 (0.01)
- O3 R O5 (0.01)
- O3 R O6 (0.01)
- O4 R O5 (0.01)
- O3 R O6 (0.01)
- O4 R O6 (0.01)
- O1 R O7 (0.01)

mobiusCapacity

{ S1 }	0
{ S2 }	0
{ S3 }	0.15
{ S1, S2 }	0.42
{ S1, S3 }	0.2
{ S2, S3 }	0.23

overallValues

O1	0.45
O2	0.5
O3	0.28
O4	0.28
O5	0.23
O6	0.23
O7	0.37
O8	0.37
O9	0.5

Not in accordance with the DM preferences

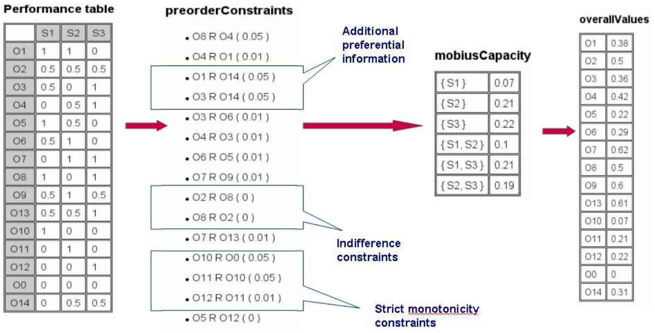
O2 = O8

Strict monotonicity required

μ{AUi} > μ{A}

Figure 3. *The first step of the aggregation/disaggregation procedure*

Ones the inconsistencies were identified, we added the corresponding constraints and we solved a new linear programming (LP). Figure 4 represents the results of this second step. The new capacity values obtained by solving (LP) provided to a new ranking of trains which was in accordance with the DMs preferences.


Performance table

	S1	S2	S3
O1	1	1	0
O2	0.5	0.5	0.5
O3	0.5	0	1
O4	0	0.5	1
O5	1	0.5	0
O6	0.5	1	0
O7	0	1	1
O8	1	0	1
O9	0.5	1	0.5
O13	0.5	0.5	1
O10	1	0	0
O11	0	1	0
O12	0	0	1
O0	0	0	0
O14	0	0.5	0.5

preorderConstraints

- O8 R O4 (0.05)
- O4 R O1 (0.01)
- O1 R O14 (0.05)
- O3 R O14 (0.05)
- O3 R O6 (0.01)
- O4 R O3 (0.01)
- O6 R O5 (0.01)
- O7 R O9 (0.01)
- O2 R O8 (0)
- O8 R O2 (0)
- O7 R O13 (0.01)
- O10 R O0 (0.05)
- O11 R O10 (0.05)
- O12 R O11 (0.01)
- O5 R O12 (0)

Additional preferential information

Indifference constraints

Strict monotonicity constraints

mobiusCapacity

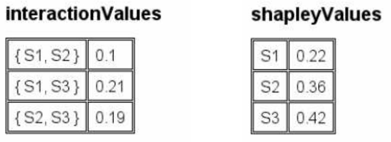
{ S1 }	0.07
{ S2 }	0.21
{ S3 }	0.22
{ S1, S2 }	0.1
{ S1, S3 }	0.21
{ S2, S3 }	0.19

overallValues

O1	0.38
O2	0.5
O3	0.36
O4	0.42
O5	0.22
O6	0.29
O7	0.62
O8	0.5
O9	0.6
O13	0.61
O10	0.07
O11	0.21
O12	0.22
O0	0
O14	0.31

Figure 4. *The second step of the aggregation/disaggregation procedure*

After identifying the capacities, one can proceed to an analysis of the preferential information by computing the *interaction Indices* I_μ and *Shapley values* Φ_μ in order to better understand the nature of interactions among criteria or to have an idea about the intensity of the importance of criteria. We can see in Fig. 5 that all interaction indices are positive which means that the criteria are rather complementary. The interaction between $S1$ and $S3$ is the most important. We can see also that the thermal comfort in the season $S3$ is the most important one and $S1$ is less important than $S2$ and $S3$. Our experts said that this analysis corresponded very well to their intuition.


interactionValues

{ S1, S2 }	0.1
{ S1, S3 }	0.21
{ S2, S3 }	0.19

shapleyValues

S1	0.22
S2	0.36
S3	0.42

Figure 5. *The interaction indices and Shapley Values*

4 Conclusion

In this paper we presented a real world application. Our application shows how we constructed the hierarchical model representing the thermal comfort in the high speed trains, how we chose the aggregation methods and how we elicited the parameters of these methods.

We wanted to point out that a real world application could need several aggregation steps and each step could require a different aggregation method. The choice of the aggregation method must be done with the DMs and experts using a guided approach.

The SNCF insisted on the fact that the results of the application must be in accordance with the perception of train passengers. We thought that for this purpose an elicitation method using some comparison examples coming from surveys with passengers is very adequate.

Sometimes the results obtained by eliciting all the parameters using some examples may provide some unexpected results. For instance if all the parameters of Electre Tri (the weights, the profiles, the thresholds) are elicited all together, one can obtain importance weights which are in contradiction with the intuition of the DMs since they depend also on the profiles. For this reason we think that if some of the parameters can be elicited directly with experts, we have to use these elicitations and then complete them by using more sophisticated methods based on comparison examples.

An aggregation/disaggregation approach (step 1 : proposing comparison examples, step 2 : using them in order to determine some parameters, step 3: presenting some new results to the DM using the results of the second step, step 4 : integrating the comments of the DMs on the step 3 in order to better determine parameters, ...) is appreciated by the DMs and the experts.

REFERENCES

[1] R. Azibi and D. Vanderpooten. Construction of rule-based assignment models. *European Journal of Operational Research*, 138(2):274–293, April 2002.

[2] C.A. Bana, E. Costa, and J.C. Vansnick. A theoretical framework for Measuring Attractiveness by a Categorical Based Evaluation TecHnique (MACBETH). In *Proc. XIth Int. Conf. on MultiCriteria Decision Making*, pages 15–24, August 1994.

[3] D. Dubois and H. Prade. Weighted minimum and maximum operations in fuzzy set theory. *Information Sciences*, 24:205–210, 1986.

[4] P.O. Fanger. *Thermal comfort*. McGraw-Hill, New York, 1973.

[5] J. Figueira and B. Roy. Determining the weights of criteria in the ELECTRE type methods with a revised Simos' procedure. *European Journal of Operational Research*, 139:317–326, 2001.

[6] M. Grabisch, I. Kojadinovic, and P. Meyer. A review of methods for capacity identification in choquet integral based multi-attribute utility theory: Applications of the Kappalab R package. *European Journal of Operational Research*, 186(2):766–785, 2008.

[7] S. Greco, B. Matarazzo, and R. Slowinski. A new rough set approach to evaluation of bankruptcy risk. In C. Zopounidis, editor, *Operational tools in the management of financial risks*, pages 121–136. Kluwer Dordrecht, Dodrecht, 1998.

[8] S. Greco, B. Matarazzo, and R. Slowinski. The use of rough sets and fuzzy sets in MCDM. In T. Gal, T. Stewart, and T. Hanne, editors, *Advances in MCDM models, Algorithms, Theory, and Applications*, pages 14.1–14.59. Kluwer Academic, Dordrecht, 1999.

[9] I. Kojadinovic. Minimum variance capacity identification. *European Journal of Operational Research*, 177(1):498–514, 2007.

[10] I. Kojadinovic. Quadratic distances for capacity and bi-capacity approximation and identification. *4OR: A Quarterly Journal of Operations Research*, 5(2):117–142, 2007.

[11] I. Kojadinovic, J-L. Marichal, and M. Roubens. An axiomatic approach to the definition of the entropy of a discrete choquet capacity. *Information Sciences*, 172:131–153, 2005.

[12] Ch. Labreuche and M. Grabisch. The choquet integral for the aggregation of interval scales in multicriteria decision making. *Fuzzy Sets and Systems*, 137:11–26, 2003.

[13] J-L. Marichal. An axiomatic approach of the discrete choquet integral as a tool to aggregate interacting criteria. *IEEE Tr. on Fuzzy Systems*, 8(6):800–807, 2000.

[14] J-L. Marichal and M. Roubens. Determination of weights of interacting criteria from a reference set. *European Journal of Operational Research*, 124:641–650, 2000.

[15] B. Mayag, M. Grabisch, and Ch. Labreuche. A representation of preferences by the choquet integral with respect to a 2-additive capacity. *Theory and Decision*, 71(3):297–324, 2011.

[16] C. Melltet and M. Hernandez. Projet ACONIT, phase 3 - Analyses des donnes : valuation des indices de confort et recherche d'interaction entre modalits, rapport d'essais AEF, tomes 1 6. Internal report, SNCF, AEF, 2006.

[17] P. Meyer and M. Roubens. Choice, ranking and sorting in fuzzy multiple criteria decision aid. In J. Figueira, S. Greco, and M. Ehrgott, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, pages 471–503. Springer Verlag, New York, 2005.

[18] V. Mousseau, J. Figueira, and J.P. Naux. Using assignment examples to infer weights for ELECTRE TRI method: Some experimental results. *European Journal of Operational Research*, 130(2):263–275, 2001.

[19] V. Mousseau and R. Slowinski. Inferring an ELECTRE TRI model from assignment examples. *Journal of Global Optimization*, 12(2):157–174, 1998.

[20] T. Murofushi and M. Sugeno. An interpretation of fuzzy measure and the choquet integral as an integral with respect to a fuzzy measure. *Fuzzy Sets and Systems*, 29:201–227, 1989.

[21] M. Mzali. Projet ACONIT, Synthse phase 2: Exploration du confort global. Internal report, SNCF, Direction de l'Innovation et de la Recherche, Novembre 2005.

[22] M. Mzali and C. Talotte. Projet ACONIT, phase 3 - Analyse des corrlations entre les mesures perceptives et mesures physiques - Volume 2 : Bilan de l'analyse des vnements de confort et du confort global et perspectives. Internal report, SNCF, Direction de l'Innovation et de la Recherche, 2009.

[23] Z. Pawlak. Rough sets. *International Journal of Computer and Information Science*, 11:341–356, 1982.

[24] B. Roy. The outranking approach and the foundations of ELECTRE methods. *Theory and Decision*, 31(1):49–74, 1991.

[25] M. Sugeno and T. Murofushi. Choquet integral as an integral form for a general class of fuzzy measures. pages 408–411, Tokyo, Japan, 1987. 2nd IFSA Congress.

[26] S. Sgrtain. Confort thermique dans les trains, Synthse du projet de recherche exploratoire. Internal report, SNCF, Direction de l'Innovation et de la Recherche, 2009.

[27] S. Sgrtain. Projet ESTHER-synthse des lot 1 et 2: Diagnostic perceptif et physique du confort thermique dans les trains. Internal report, SNCF, Direction de l'Innovation et de la Recherche, 2011.

[28] A. Ngo The and V. Mousseau. Using assignment examples to infer category limits for the ELECTRE TRI method. *Journal of Multi-Criteria Decision Analysis*, 11(1):29–43, 2002.

[29] R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Transactions on Systems, Man and Cybernetics*, 18(1):183–190, 1988.

Dynamic management and learning of user preferences in a content-based recommender system

Lucas Marin[1], **Antonio Moreno**, **David Isern** and **Aida Valls**

Abstract. One of the most challenging tasks in the construction of recommender systems is the definition of mechanisms that permit to automatically learn the user's interests and combine different types of data. This paper presents a framework that stores the preferences of the user on a set of numerical and categorical criteria. The system is able to analyse the selections of the user and dynamically adapt his/her preferences over time. The management of the information is domain independent, since the framework can be applied to any scenario in which the user is constantly faced with a decision problem. A study of the performance of these novel adaptation techniques shows promising results.

1 Introduction

In the current *Knowledge Society* users have access to a huge number of information, data and knowledge sources. Due to the astonishing speed at which new content is created and published, evaluating all the incoming information has become a difficult, time-consuming and overwhelming task. In this scenario, users are constantly confronted with situations in which they are receiving a continuous stream of data and they have to find the items that fit better with their needs and preferences. That is the reason why the necessity of using a *Recommender System* (RS) to assist these tasks is increasing every day [3,8,19].

The work described in this paper[2] assumes a context in which alternatives are described by both numeric and categorical attributes. The set of alternatives contains the proposals available in the RS such as a tourist destinations, films, restaurants, etc. The alternatives are described by means of a set of criteria that can be independently evaluated. For instance, if the alternatives are restaurants then the criteria could be related to the type of food, the average price and the distance to the city centre. In this case, each proposal known by the RS would be defined with two numerical values (average price, distance to the city centre), and a categorical one (the type of food, that can be Mediterranean, Indian, Chinese, etc.).

As noted by several researchers, the definition of the degree of expressivity of the preferences can be a hard problem [1,14,16]. In most cases users find it very difficult to express their preferences using a numerical scale. So, a linguistic approach can be employed to handle preferences using linguistic terms, providing a higher modelling flexibility [16].

In this work the user's preferences over the attribute values are given in linguistic terms, although a transformation from the numeric preference function is necessary in the case of numeric attributes (see section 7). The main goal is to design algorithms capable of automatically learning the user's preferences from his/her interaction with the system, without the user having to spend time making an explicit declaration on his/her preferences on all the values of all the attributes. These algorithms will be especially suitable in situations in which the user is constantly confronted with a decision problem (e.g., which news to read every morning), rather than in single-shot (or very infrequent) decisional problems. In a setting in which the user has to make choices very often, the user profile adaptation algorithms will benefit from them to learn quickly the user's interests.

The rest of the paper is organised as follows. Section 2 makes a very brief overview of the area. Section 3 describes how each of the alternatives of a decision problem is evaluated, taking into account the current information in the user profile. Section 4 shows the whole architecture of the recommender system, including the module of preference adaptation. Sections 5 explains how the system learns the preferences over categorical attributes, whereas section 6 extends the learning mechanisms to the case in which the categorical attribute is multi-valued. Section 7 describes the adaptation mechanisms to learn numeric preferences. Section 8 explains the evaluation procedure of the adaptation techniques and comments the system's performance. Finally, some conclusions and future lines of work are presented.

2 Background

As introduced by [17], the main problem of recommender systems is the incompleteness in the representation of the user's preferences. In many cases the adopted solution is the simplification of the models of the users and the alternatives. Some of the initial attempts to tackle this problem were the construction of utility functions (based on *multi-attribute utility theory* (MAUT)). These types of systems have high confidence on the results because those utility functions are specially designed on

[1] ITAKA (Intelligent Technologies for Advanced Knowledge Acquisition) research group. Department of Computer Science and Mathematics, Universitat Rovira i Virgili, Avinguda Paisos Catalans, 26, 43007 Tarragona, Catalonia (Spain). Email addresses: lucas.marin@urv.cat, antonio.moreno@urv.cat, david.isern@urv.cat, aida.valls@urv.cat. Website: http://deim.urv.cat/~itaka.

[2] Some of this work has already been presented in previous papers from the authors [7,10,11,12,13].

each problem / domain [2] but, at the same time, they have a low level of generalization.

Another approach to deal with alternatives is to sort and classify them according to a set of attributes. [15] propose the ELECTRE TRI model that permits the decision maker to dynamically classify (even hierarchically) the set of alternatives. However, this model is based on optimizations designed for problems in which the user interests do not change over time.

Differently, in recommender systems it is more appropriate to take an adaptive and constructive approach to the decision task, such as the model of critiquing-based RS [4,5]. This approach uses conversational models (using natural language techniques) and graphical user interfaces to guide the users to efficiently target their ideal products and learn from the user feedback.

Recently, query-based systems have been proposed for preference elicitation [2,20]. Queries are commonly used in conjoint analysis and product design [9], requiring a user to indicate which alternative is most preferred from a set of options. Under some very general assumptions, optimization of choice queries reduces to the simpler problem of choosing the *optimal recommendation set*, so that if a user were forced to choose one, it maximizes the expected utility. Our work is partially based on this model, assuming that the selection of one alternative from the set is representative enough to extract some knowledge about the user's preferences.

3 Linguistic aggregation operators

This section describes how the preference information that is used by the RS is represented in the user profile. When the RS is evaluating the overall preference score of a given alternative, the information in the user profile is used to know the partial preference score for each criterion, which is given in a common linguistic scale (such as "High", "Medium" or "Low"). After that, all the linguistic scores are aggregated to obtain an overall score for the alternative. Although the RS is based on linguistic scales, the model of preferences is different depending on the nature of the attribute that is being evaluated, distinguishing the numerical measurement scales from the categorical ones.

In the case of numerical attributes, the user profile stores a numerical preference transformation function. Afterwards, a translation into a linguistic term can be made. In the case of categorical attributes, the user profile represents directly the degree of preference over each possible value using the linguistic scale, selecting the appropriate term "High", "Medium", "Low", etc.

The definition of the meaning of those terms is made by using fuzzy sets as can be seen in figure 1, where 5 fuzzy linguistic preference terms are defined.

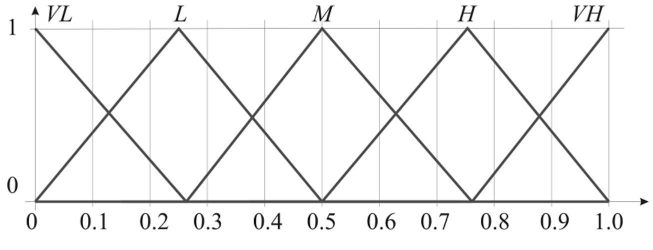


Figure 1. Example of a balanced linguistic preference set

Terms in Figure 1 are distributed uniformly across the 0-1 domain: all the fuzzy sets have the same shape and are symmetrical. When it is necessary to aggregate some linguistic terms into a single value, the Linguistic Ordered Weighted Averaging (LOWA, [6]) aggregation operator is usually employed. However, there may be cases in which it is necessary to add expressivity to the definition of the linguistic terms, as there are problems in which it is useful to associate to the linguistic terms fuzzy sets that have different and irregular shapes, such as those shown in figure 2. Aggregating terms in those irregular sets, known as *unbalanced* term sets, requires a more complex aggregation procedure.

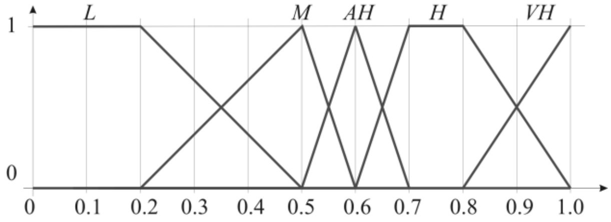


Figure 2. Example of an unbalanced linguistic preference set

For this purpose, we designed a new aggregation operator called ULOWA (*Unbalanced LOWA*) which enables the aggregation of fuzzy terms defined on an unbalanced fuzzy set. This operator is defined in [7] as follows:

$$ULOWA(a_1,...,a_n) = W \cdot B^T = C^n\{w_k, b_k, k = 1,..,n\} =$$
$$= w_1 \otimes b_1 \oplus (1-w_1) \otimes C^{n-1}\{\beta_h, b_h, h = 2,..,n\}$$

In this expression W is the set of weights, $\beta = w_h / \sum^n{}_2 w_h, h = \{2,\ldots,n\}$ and $B = \{b1,\ldots,b_n\}$ is a permutation of the elements of A, such that B=σ(A)=$\{a_{\sigma(1)},\ldots, a_{\sigma(n)}\}$, where $a_{\sigma(j)} \leq a_{\sigma(i)}$, $\forall\, j \leq i$. C^n is the convex combination operator of n labels. If $\exists j \in 1..n$, $w_j=1$ and $\forall k \in 1..n$, $k \neq j$, $w_k=0$, then $C^n\{w_i,b_i,i=1,..,n\}=b_j$. When $n=2$, with $b_1=s_j$ and $b_2=s_i$, $s_j,s_i \in S(j \geq i)$ then

$$C^2\{w_i, b_i, i = 1,2\} = w_1 \otimes s_j \oplus (1-w_1) \otimes s_i = s_k$$

such that $k=\text{argmax}_{i \leq p \leq j}\{Sim(s_p,\delta)\}$, where δ is an intermediate crisp number between s_j and s_i defined as $\delta = (x_k,x_k,x_k,x_k)$, with $x_k = x^*{}_{si} + w_1(x^*{}_{sj} - x^*{}_{si})$, $x^*{}_s$ being the x-component of the Center of Gravity of the label S. Sim(P,Q) calculates the similarity between two fuzzy numbers P and Q with the formula

$$Sim(P,Q) = \sqrt[4]{\prod_{i=1}^{4} \; 2 - |p_i - q_i| \;} - 1$$

where each triangular or trapezoidal fuzzy set is represented in the usual way by four numbers.

This work was later expanded in the definition of the IULOWA aggregation operator (*Induced ULOWA*) which supports the induction of the order of the arguments by taking into account the measures of fuzziness and specificity of the aggregating terms [13].

4 Multi-criteria recommender system

Many points of view or criteria define the alternatives that the RS aims to recommend. Those criteria can be categorical (*e.g.*, the attribute "Climate" when defining a "Tourist destination" alternative) or numerical (*e.g.* the criteria "Population density" for the same case).

The steps followed by the designed RS in order to give a recommendation to the user are as follows:

1. Identify the set of possible alternatives that can be recommended to the user.
2. Obtain a linguistic evaluation of each alternative. This is done by the following procedure:
   a) Obtain a numeric preference over the value of each numeric attribute by using the numeric preference function defined in the user profile (more details on section 7).
   b) Translate the numeric preference value to a linguistic one by mapping it in the linguistic fuzzy term domain.
   c) Obtain the value of preference over the value of each categorical attribute by observing the associated linguistic label in the user profile.
   d) Aggregate all the linguistic terms using the ULOWA aggregation operator.
3. Sort all the alternatives by descending order according to the evaluation obtained through the aggregator.

The first ranked alternatives are the ones that fit better the user interests, according to the information in the user profile. If the user selects an alternative that is not in the first position, it means the user preferences stored in the profile are not accurate enough and the preference learning algorithm explained on further sections comes into play. A graphical representation of the whole process can be seen in figure 3 .

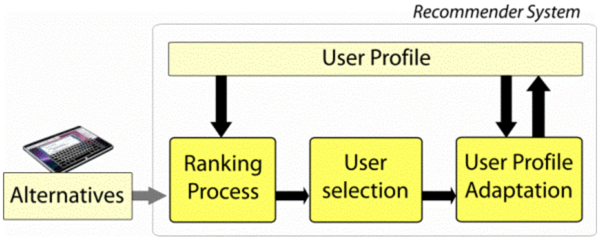


Figure 3. Architecture of the recommender framework

5 Learning of linguistic preferences over categorical criteria

As previously stated, preferences over categorical attributes are represented using a linguistic term scale, in which each term has an associated fuzzy set. Concretely, in the user profile each categorical value has a linguistic term of preference associated to it. To learn and adapt those values of preference, it is first necessary to identify when the RS has not been accurate enough. That occurs, as indicated above, when the user selects an alternative that is not the first one ranked by the RS. In this case, two pieces of feedback are extracted: the alternatives ranked above the selection (which are called the *over ranked alternatives*) and the selection itself.

The main idea is to find attribute values repeated among the over ranked alternatives that do not appear on the selection, which will be the candidates for having his preference decreased. Similarly, the preference of the attribute values that appear on the selection and do not appear often on the over ranked alternatives is likely to be increased. The interested reader may find a more detailed explanation of the process of adaptation of linguistic preferences in [12].

The profile adaptation is conducted by two processes. The first one—called *on-line adaptation*—is executed every time the user asks the system for a recommendation, and it evaluates the information that can be extracted from the current ranked set of alternatives. The main goals of this stage are to decrease the preference of the attribute values that are causing non-desired alternatives to be given high scores and to increase the preference of the attribute values that are important for the user but are not well judged on the basis of the current user profile. For each recommendation made by the system, two sources of information are evaluated: the selected alternative, which is the choice made by the user, and the alternatives that were ranked above it. Values extracted from the over ranked alternatives have their level of preference decreased whereas the ones extracted from the user's final selection that do not appear in the set of over-ranked alternatives have their preference increased.

The second one—called *off-line adaptation*—is triggered after the recommender system has been used a certain number of times. It considers the information given by the history of the previous rankings of alternatives and the selections made by the user in each case, but considers that information separately. When the system faces cases in which the number of over ranked alternatives is not large enough for reliable conclusions to be extracted, it stores the small number of over ranked alternatives in a temporary buffer. After several iterations in which the number of over ranked alternatives has been insufficient for evaluation, the system will have recorded enough alternatives to start evaluating them. When there are enough saved over ranked alternatives, the values in their attributes will be analysed and their preference decreased. Moreover, user selections are also stored, and after a certain number of choices have been made, they are evaluated with the objective to increase the preference of the most repeated attribute values, since their repeated selection indicates that the user is really interested in them.

6 Management of multi-valued categorical criteria

The previous section considered the case in which categorical attributes could only take one single value. However there are cases in which many values for the same attribute may appear. An example can be the attribute "*Types of food*" in a restaurant. If a user has a "*High*" preference over "*Asian food*" restaurants and a "*Low*" preference over "*Rice dishes*", we can argue that the preference we could assign to the "*Type of food*" attribute in a restaurant with both values should be "*Medium*" (an average of the two kinds). If another restaurant only offers "*Asian food*" then its preference should be "*High*", so this restaurant would have a higher ranking than the first one. The rationale of this procedure is that it seems more adequate to reward the alternatives that are more focused on the aspects the user really likes. This example represents an "average" preference aggregation policy; however, other policies could also be considered depending on the meaning of the attributes in a particular application.

Since we cannot be sure which of the values listed in the attribute are the ones of interest for the user, it has been necessary to design a "*relevance function*" which indicates how relevant is a value found among the over ranked alternatives or in the selected alternative. Relevance is measured in a [0, 1] scale, with 1 meaning maximum relevance. To calculate how relevant a term *t* of the attribute *j* is among the over ranked alternatives we use this expression (the relevance value is 0 if it does not appear in the over ranked alternatives):

$$R_j^o(t) = \frac{1}{no}\sum_{i=1}^{nt}\frac{1}{nv_j^i} \tag{1}$$

Here, *no* represents the number of over ranked alternatives, *nt* the number of over ranked alternatives where *t* appears, and nv^i_j the number of values that appear for the attribute *j* in the alternative *i*. In this equation we consider that every linguistic term that appears in the over ranked alternatives has a relevance which is inversely proportional to the number of other values for the same attribute that appear among the entire set of over ranked alternatives.
To calculate the relevance of a term in the selection we use:

$$R_j^s(t) = \frac{1}{2}\left(\frac{1}{nv_j} + \frac{nl}{tv}\right) \tag{2}$$

Here nv_j represents the number of values that appear for the attribute *j* in the selection, *nl* the total number of linguistic attributes, and *tv* the total number of linguistic values that appear in the selection. The relevance of a term in the selection is the mean between the importance of the term among the values that appear with it in the same attribute and the importance of each linguistic term that appears in the selection compared with the number of linguistic attributes.

Finally, after calculating both partial relevancies for all the terms, the overall relevance $R_j(t)$ is calculated as:

$$R_j(t) = R_j^s(t) - R_j^o(t) \tag{3}$$

In conclusion, considering a threshold γ to avoid making changes in the profile with low relevance, it can be deduced that:

- If *Rj*(*t*)>γ, the preference over term *t* for the attribute *j* needs to be increased (moved to the next term).
- If *Rj*(*t*)<γ, the preference over term *t* for the attribute *j* needs to be decreased (moved to the previous term).

7 Learning preference functions over numerical criteria

Preferences over numeric attributes are represented with a numeric preference function as the one represented graphically on figure 4.

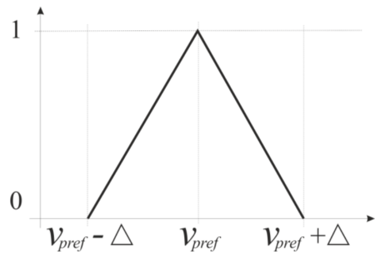


Figure 4. Basic numeric preference function

The Δ value (or width) was considered to be a 10% of the domain of the numeric variable and v_{pref} is the value of maximum preference for the user. The preference *p* over a numeric value *x* of the attribute *a* is calculated using Eq. (4). So, the task of the numeric learning algorithm in this basic case is to learn the correct v_{pref} value for the user.

$$p_a(x) = 1 - \frac{\left|x - v_{pref}\right|}{\Delta} \tag{4}$$

The numeric adaptation of the user profile is inspired by Coulomb's Law: "the magnitude of the electrostatic force of interaction between *two point charges is directly proportional to the scalar multiplication of the magnitudes of charges and inversely proportional to the square of the distance between them*".

The main idea is to consider the value stored in the profile (current preference) as a charge with the same polarity as the values of the same criterion on the over ranked alternatives, and with opposite polarity to the value of that criterion in the selected alternative [11]. Thus, the value of the profile is pushed away by the values in the over ranked alternatives and pulled back by the value in the selected alternative. Two stages have been considered in the adaptation algorithm. The first one, called *on-line* adaptation process, is performed each time the user asks for a recommendation. The other stage, called *off-line* process, is performed after a certain amount of interactions with the user.

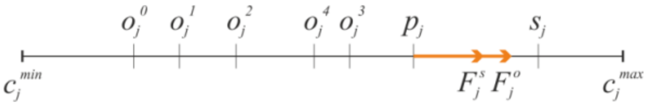


Figure 5. Attraction and repulsion forces

For the *on-line* stage, the information available in each iteration is the user selection and the set of over ranked alternatives. In order to calculate the change of the value of preference in the user profile for each criterion it is necessary to study the attraction force done by the selected alternative and the repulsion forces done by the over ranked ones in each criterion, as represented in the example in figure 5, in which the *j*-th value of the five over ranked alternatives o^0, o^1, o^2, o^3, and o^4 causes a repulsion force F^o_j, and the value for the same criterion of the selected alternative, *sj*, causes an attraction force F^s_j. Both forces are applied on the *j*-th value of the profile, *Pj*.

The attraction force *Fs* done by the selected alternative for each attribute *j* is defined as:

$$F^s \;\; P,s,j,\alpha \;\; = \begin{cases} \Delta_j \left(\dfrac{1}{\left|s_j - P_j\right|^{\alpha}} \cdot \dfrac{s_j - P_j}{\left|s_j - P_j\right|} \right) & if\ s_j \neq P_j \\ 0 & if\ s_j = P_j \end{cases} \tag{5}$$

In this equation, Δ_j is the range of the criterion *j*, s_j is the value of the criterion *j* in the selected alternative and P_j is the value of the same criterion in the stored profile *P*. The parameter *α* adjusts the strength of the force in order to have a balanced adaptation process. The repulsion force exerted by the over ranked alternatives for each criterion *j* is defined as a generalization of Eq.(5) as follows:

$$F^o \;\; P,\; o^1,\ldots,o^{no} \;\; ,j,\alpha \;\; = \sum_{i=1}^{no} \frac{1}{\left|Pj - o^i_j\right|^{\alpha}} \cdot \frac{Pj - o^i_j}{\left|Pj - o^i_j\right|} \tag{6}$$

Finally, both forces are summed up and the resulting force is calculated.

The techniques designed for the on-line stage fail at detecting user trends over time since they only have information of a single selection. The *off-line* adaptation process gathers information from several user interactions. This technique allows considering changes in the profile that have a higher reliability than those proposed by the on-line adaptation process, because they are supported by a larger set of data.

The off-line adaptation process can be triggered in two ways: the first one evaluates the user choices, while the second one analyses the over ranked alternatives discarded by the user in several iterations. The possibility of running the off-line process (in any of its two possible forms) is checked after each recommendation. In the first case, the system has collected some alternatives selected by the user in several recommendation steps, and it calculates the attraction forces (*F’s*) exerted by each of the stored selected alternatives over the values stored in the profile, using an adaptation of Eq. (5), that has as inputs the profile *P*, the past selections {s^1,…,s^{rs}}, the criterion to evaluate *j*, and the strength-adjusting parameter *α*:

$$F^{'}_s \;\; P,\; s^1,\ldots,s^{rs} \;\; ,j,\alpha \;\; = \sum_{i=1}^{rs} \frac{1}{\left|s^i_j - P_j\right|^{\alpha}} \cdot \frac{s^i_j - P_j}{\left|s^i_j - P_j\right|} \tag{7}$$

The second kind of off-line adaptation process evaluates the set of over ranked alternatives that have been collected through several iterations and which were not used in the on-line adaptation process (because it did not have enough over ranked alternatives in a single iteration). When the stored over ranked alternatives reach a certain number, the off-line adaptation process calculates the repulsion forces over the profile values exerted by those alternatives (F_o), which are calculated with Eq. (6).

This process, however, just adapts the value of maximum preference of the numeric preference function. Recently we have studied how to adapt more parameters associated to the preference function and not just that value. The new learning method relies on historic data about the user selections to approximate a more expressive preference function of the numeric attributes. With this approach, we have a new definition of the function of preference which now has 5 parameters (left and right slope, left and right width, and value of preference) instead of just the value of preference. The graphical representation of a preference function can be seen in the example in figure 6, where the left slope is a value under 1, the right slope is a value over 1, and the left width is greater than the right one.

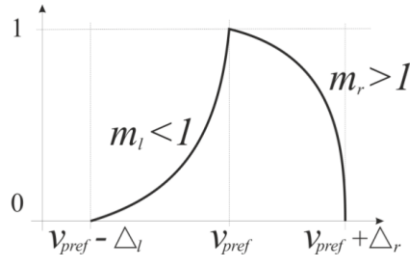


Figure 6. Numeric preference function with 5 parameters

With this new definition of the preference function, the numerical preference associated to a particular value *x* of attribute *a* is calculated as follows:

$$p_a(x) \begin{cases} 1 - \dfrac{\left|x - v_{pref}\right|^{m_l}}{\Delta_l} & if\,(x < v_{pref}) \\ 1 & if\,(x = v_{pref}) \\ 1 - \dfrac{\left|x - v_{pref}\right|^{m_r}}{\Delta_r} & if\,(x > v_{pref}) \end{cases} \tag{8}$$

In this expression m_l and m_r are the function slope values (for the left and right sides of the function) and Δ_l and Δ_r are the parameters which define the width of the function (also for the left and right sides).

In order to learn all those new parameters, the first step is to obtain the more reliable values from the historic set of selections. This is done by extracting a percentage of the values closer to the value of preference (trust interval), normally of 90%. This filter is useful to get rid of outlier values. Then a probability distribution function, represented with a histogram, is calculated with those values.

The sample or discretization step is a parameter, normally around 1% of the domain range. Delta values are then calculated by observing the width of the probability distribution. For example, if the first value different to 0 in the histogram is 3 and the last is 56, and the value of higher preference (v_{pref}) is 34, Δ_l would be 31 and Δ_r would be 22.

Afterwards, the algorithm generates preference functions with different combinations of values for the slope values (*m*) (in the range from 0 to 4 in steps of 0.2), and compares the distance between each preference function and the probability distribution. The function with the lower distance determines the chosen slope.

Finally, the new preference function is built with the new delta and slope values.

8 Performance of the adaptation algorithms

The performance of the preference learning algorithms has been tested in several domains such as tourist destinations, news articles (obtained from The Times[3] newspaper) and restaurants (obtained from BCN Restaurantes[4]) [10,12]. The evaluation shown in this section has been done using a data file containing 3000 restaurants that were divided in groups of 15, giving a total of 200 different recommendations. The attributes considered were "Type of food" (multi-valued categorical with 15 possible values), "Atmosphere" (multi-valued categorical with 14 possible values), "Special characteristics" (multi-valued categorical with 12 possible values), "Average price" (numerical in a 20 to 60 € range) and "Distance to city centre"(numerical in a 0 to 5 km domain).

Three random initial profiles were generated and an ideal profile was created manually. The objective is that, after the 200 recommendations, the three random profiles are close to the ideal one by learning the user preferences through the analysis of the chosen (and over ranked) alternatives in each iteration. The evaluation process, executed independently for each initial profile, has the following steps, which are repeated 200 times:

1) Select the next block of 15 alternatives.
2) Rank those alternatives using the current user profile.
3) Rank the alternatives using the ideal profile. The option that would be chosen by the user is the first one in this rank.
4) Look for the position of the selection obtained in the previous step in the ranked list obtained in step 2 (*i.e.*, simulate the user selection). The alternatives that are ranked above this item are considered over ranked and treated as feedback for the learning process.
5) Identify the possible changes to the profile and perform the ones with greater reliability.

At each iteration of the process, the distance between the current profile and the ideal one is calculated using a measure designed for this purpose. This measure gives a result in the [0,1] domain, 0 meaning that both profiles are identical (optimal case) and 1 meaning that both profiles are completely opposed.

The distance between numeric attributes is calculated as

$$d(n,c,i) = 1 - p_n^c(v_{pref\,n}^{\;i}) \tag{9}$$

where n is the numerical attribute, c is the current profile (the one being learned), i is the ideal profile, and $p_n^c(v_{pref\,n}^{\;i})$ is the value of preference of the v_{pref} value for the attribute n in i using the preference function of the same attribute in the profile c. A distance 0 means that the v_{pref} values in both profiles are equal.

The equation to calculate the distance between categorical attributes is

$$d(l,c,i) = \frac{1}{card(l)} \sum_{k=1}^{card(l)} \frac{\left|CoG(p_l^c(v_k)) - CoG(p_l^i(v_k))\right|}{\left|CoG(s_{\min}) - CoG(s_{\max})\right|} \tag{10}$$

where l is the categorical attribute, *card(l)* is the cardinality of the attribute l (i.e., the number of different linguistic values it can take), $CoG(p_l^c(v_k))$ and $CoG(p_l^i(v_k))$ are the x-coordinate of the centres of gravity of the fuzzy linguistic labels associated to the value of preference of v_k in the profiles c and i, respectively, and $CoG(s_{\min})$ and $CoG(s_{\max})$ are the centres of gravity of the minimum and maximum labels of the domain, respectively. Finally, the distance between two profiles is calculated as

$$D(c,i) = \frac{1}{na} \sum_{k=1}^{na} d(k,c,i) \tag{11}$$

where na is the total number of attributes.

The average evolution of that distance can be seen in figure 7. As it can be observed, the distance between profiles decreases to 0.2 after 50 recommendations. This performance can be better understood by observing figure 8, which represents the position of the alternative the user selects in the ranked set of alternatives (the lower the better). After about 50 iterations, the selected alternative is among the first three ones in 95% of the cases (and the first one in around 70% of the cases).

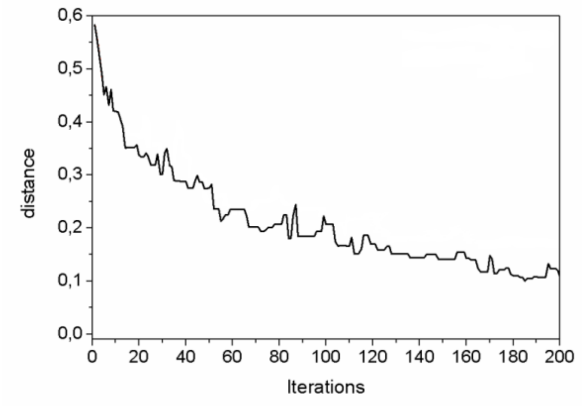


Figure 7. Distance evolution

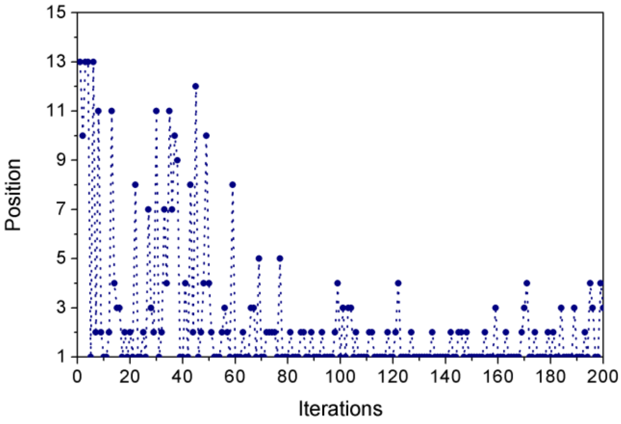


Figure 8. Position of the selected alternative

[3] Website: http://www.thetimes.co.uk/tto/news/

[4] Website: http://www.bcnrestaurantes.com/eng/

The interested reader can find in [12] a more detailed account of the influence of the different parameters of the adaptation algorithms in the final result.

9 Conclusions and future lines of research

The main objective of this paper is to present different techniques of profile learning to enable a Recommender System (RS) to automatically and dynamically adapt the user's preferences to increase the accuracy of the recommendations. The alternatives (or set of possible solutions to the recommendation problem) are defined by multiple criteria that can be either numerical or categorical. Categorical attributes can be multi-valued.

The algorithm combines the use of past and recent information to adapt the current user's profile. This combination is set up with different parameters that permit to balance the adaptation towards past information or more recent one. As shown in the last section, the algorithm converges as the profile evolves over time. Some recent tests, not reported in this paper, also show a high confidence of the results when changes of the ideal profile are considered during the simulation.

As more information has the algorithm, more and faster patterns can be inferred and the algorithm obtains more accurate results. As shown through the paper, the current proposal is a content-based one taking into account data collected from a single user. A next step is to use information from other users turning the algorithm into a collaborative-based one. If we can compare two different users, a possibility is to use selections made by one user and also combine them with selections made by similar users. With this approach, the learning curve of the user's profile could potentially be diminished, although accuracy problems could appear as noticed in [18].

Acknowledgements

This work has been supported by the Universitat Rovira i Virgili (a pre-doctoral grant of L. Marin) and the Spanish Ministry of Science and Innovation (DAMASK project, *Data mining algorithms with semantic knowledg*e, TIN2009-11005) and the Spanish Government (Plan E, Spanish Economy and Employment Stimulation Plan).

References

[1] G. Adomavicius, A. Tuzhilin, Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions, *IEEE Transactions on Knowledge and Data Engineering* **17** (6) (2005), 734-749.

[2] D. Braziunas, C. Boutilier, Elicitation of Factored Utilities, *AI Magazine* **29** (4) (2008), 79-92.

[3] R. Burke, Hybrid Recommender Systems: Survey and Experiments, *User Modeling and User-Adapted Interaction* **12** (4) (2002), 331-370.

[4] L. Chen, P. Pu, Interaction design guidelines on critiquing-based recommender systems, *User Modeling and User-Adapted Interaction* **19** (3) (2009), 167-206.

[5] K. Gajos, D. S. Weld, Preference elicitation for interface optimization, in: *18th annual ACM symposium on User interface software and technology, UIST 2005*, P. Baudisch, M. Czerwinski, eds, ACM New York, Seattle, Washington, USA, 2005, 173-182.

[6] F. Herrera, J. L. Verdegay, Linguistic assessments in group decision, in: *First European Congress on Fuzzy and Intelligent Technologies, EUFIT 1993*, Aachen, Germany, 1993, 941-948.

[7] D. Isern, L. Marin, A. Valls, A. Moreno, The Unbalanced Linguistic Ordered Weighted Averaging Operator, in: *IEEE World Congress on Computational Intelligence, WCCI 2010*, IEEE Computer Society, Barcelona, Catalonia, 2010, 3063-3070.

[8] G. Linden, B. Smith, J. York, Amazon.com recommendations: Item-to-item collaborative filtering, *IEEE Internet Computing* **7** (1) (2003), 76-80.

[9] J. J. Louviere, D. A. Hensher, J. D. Swait. *Stated choice methods: analysis and application*, Cambridge University Press, 2000.

[10] L. Marin, A. Moreno, D. Isern, Automatic learning of preferences in numeric criteria, in: *Catalan Conference on Artificial Intelligence, CCIA 2011*, C. Fernandez, H. Geffner, F. Manya, eds, IOS Press, Lleida, 2011, 120-129.

[11] L. Marin, A. Moreno, D. Isern, Preference Function Learning over Numeric and Multi-valued Categorical Attributes, in: *Workshop on Preference Learning: problems and applications in AI in conjunction with ECAI 2012*, J. Fürnkranz, E. Hüllermeier, eds, Montepellier, France, 2012, 36-41.

[12] L. Marin, D. Isern, A. Moreno, A. Valls, On-line dynamic adaptation of fuzzy preferences, *Information Sciences* **220** (2013), 5-21.

[13] L. Marin, J. M. Merigó, A. Valls, A. Moreno, D. Isern, Induced Unbalanced Linguistic Ordered Weighted Average, in: *7th conference of the European Society for Fuzzy Logic and Technology (EUSFLAT-2011) and LFA-2011*, S. Galichet, J. Montero, G. Mauris, eds, Atlantis Press, Aix-les-Bains, France, 2011, 1-8.

[14] M. Montaner, B. López, J. L. de La Rosa, A taxonomy of recommender agents on the internet, *Artificial Intelligence Review* **19** (4) (2003), 285-330.

[15] V. Mousseau, R. Slowinski, Inferring an ELECTRE TRI model from assignment examples, *Journal of global optimization* **12** (2) (1998), 157-174.

[16] C. Porcel, A. G. López-Herrera, E. Herrera-Viedma, A recommender system for research resources based on fuzzy linguistic modeling, *Expert Systems with Applications* **36** (3, Part 1) (2009), 5173-5183.

[17] R. Price, P. R. Messinger, Optimal Recommendation Sets: Covering Uncertainty over User Preferences, in: *20th national conference on Artificial intelligence, AAAI 2005*, A. Cohn, ed, AAAI Press, Pittsburgh, Pennsylvania, USA, 2005, 541-548

[18] L. Quijano-Sánchez, J. A. Recio-García, B. Díaz-Agudo, G. Jiménez-Díaz, Social Factors in Group Recommender Systems, *ACM Transactions on Intelligent Systems and Technology* **in press** (2012).

[19] P. Resnick, H. R. Varian, Recommender Systems, *Communications of the ACM* **40** (3) (1997), 56-58.

[20] P. Viappiani, C. Boutilier, Optimal Bayesian Recommendation Sets and Myopically Optimal Choice Query Sets, in: *Twenty-Fourth Annual Conference on Neural Information Processing Systems, NIPS 2010*, Vancouver, Canada, 2010, 2352-2360.

An algorithm for active learning of lexicographic preferences

Fabien Delecroix[1], Maxime Morge[2], Jean-Christophe Routier[3]

Abstract. At the crossroad of preference learning and multicriteria decision aiding, recent researchs on preference elicitation provide useful methods for recommendation systems. In this paper, we consider (partial) lexicographic preferences. In this way, we can consider dilemmas and we show that these situations have a minor impact in practical cases. Based on this observation, we propose an algorithm for active learning of preferences. This algorithm solve the dilemmas by suggesting concrete alternatives which must be ranked by the user.

1 Introduction

At the crossroad of preference learning and multicriteria decision aiding, recent researchs on preference elicitation provide useful methods for recommendation systems [5]. Recommender systems are aimed at helping users to deal with the problem of information overload by facilitating access to relevant items [8] (web resources, products, services, ...). These systems attempt to generate a model of the user or user's task and apply different heuristics to anticipate what item may be of interest to the user. Recommender systems can be collaborative, which build on similarities between users with respect to the items, or content-based, which build on similarities between items that the user liked in the past. However, these approaches make the assumption that we have prior information about the user and they focus on how to use this information rather than how to get it [6]. By contrast, we consider in our approach not having such data on user but we aim at collecting proactively these data [3, 2]. In this perspective, the main issue is the following one: "which question(s) to ask to a decision-maker in order to identify a relevant item in a user-friendly way?"

In this paper, we consider active learning of preferences (see [9]). We aim at inferring the preferred alternative(s) from selected observed situations. We focus here on multi-attribute decision making where alternatives are mutually exclusive and attributes are independent. Based on an ordinal approach of preferences, we are taking into account situations of dilemmas. For this purpose, we consider lexicographic preference orders which are partial. This model allows us to express the relative importance of attributes and the dilemmas due to the partial nature of preferences. In this paper, we show that these dilemmas have a minor impact in practical cases. Based on this observation, we propose an algorithm for active learning. This algorithm solve the dilemmas by suggesting concrete alternatives which must be ranked by the user.

The paper is organized as follows : we first introduce the notions of preferences and choices in the background of this work (cf Section 2) and we describe a multi-attribute decision-making problem using a partial lexicographic order in Section 3. Section 4 evaluates the impact of dilemmas due to the incomparabilities. Then, we propose an active algorithm for learning preferences which aims at identifying a preferred alternative (cf Section 5). Section 6 discusses some related works. Section 7 concludes with some directions for future works.

2 Decision

Decision making is the cognitive process of selecting a plan of action based on preferences. In a decision problem, the goal of the decision maker is to choose within a set of alternatives, the ones which maximize the satisfaction of the decision-maker.

2.1 Preferences

In a decision problem, the decision-making must choose between some alternatives. We suppose here that there is a preference relation capturing the penchant of the decision-maker which allows to compare the alternatives. We formalize here some well-known relations of preference.

Notation 1 (Relation of preference). *Let* $\texttt{Alt}$ *be a set of alternatives. We denote* $\succsim \ \subseteq \texttt{Alt} \times \texttt{Alt}$ *the (weak) relation of preference over* $\texttt{Alt}$.
If $x \succsim y$, *we say that "the alternative* x *is at least as good as* y*".*

Each alternative is at least as good at itself.

Axiom 1. *The weak relation of preference* $\succsim$ *over* $\texttt{Alt}$ *is reflexive, i.e.* $\forall\, x \in \texttt{Alt},\ x \succsim x$.

Axiom 2. *The weak relation of preference* $\succsim$ *is transitive, i.e.* $\forall\,(x, y, z) \in \texttt{Alt}^3, x \succsim y$ *and* $y \succsim z \Rightarrow x \succsim z$.

According to the axioms 1 and 2, a weak relation of preference (cf. Notation 1) is a preorder (reflexive and transitive).

We can define the strong relation of preference and the equivalence relation (called indifference) from the weak relation of preferences.

Definition 1 (Strong relation of preference). *Let* $\texttt{Alt}$ *be a set of alternatives and* $\succsim$ *a weak relation of preference over* $\texttt{Alt}$. *The strong relation of preference* $\succ \ \subseteq \texttt{Alt} \times \texttt{Alt}$ *is defined such that:* $\forall (x,\ y) \in \texttt{Alt}^2,\ x \succ y \Leftrightarrow x \succsim y$ *and* $\neg\,(y \succsim x)$.
If $x \succ y$, *we say that "*x *is strongly preferred to* y*"*

Remark 1. *Contrary to the weak relation of preference which is reflexive (cf. Axiom 1), a strong relation of preference is irreflexive and so, it is not a preorder.*

[1] Université Lille 1, Fabien.Delecroix@lifl.fr
[2] Université Lille 1, Maxime.Morge@univ-lille1.fr
[3] Université Lille 1, Jean-Christophe.Routier@univ-lille1.fr

From the weak relation of preference, we can define a relation of indifference in order to capture the insensibility of the decision-maker with respect to some alternatives.

Definition 2 (Relation of indifference)**.** *Let* $\texttt{Alt}$ *be a set of alternatives and* $\succsim$ *a weak relation of preference over* $\texttt{Alt}$*. We define the relation of indifference* $\sim \ \subseteq \ \texttt{Alt} \times \texttt{Alt}$ *such that:* $\forall(x,\ y) \in \texttt{Alt}^2,\ x \sim y \Leftrightarrow x \succsim y$ *and* $y \succsim x$*.*
If $x \sim y$*, we say that "the decision-maker is indifferent between* x *and* y*".*

Since we consider the indifference as the absence of strong preference, $x \sim y$ means that the decision-maker feels that, in a preference sense, there is no real difference between x and y.

Remark 2. *The relation of indifference is a relation of equivalence, i.e. reflexive, symmetric and transitive.*

From the weak relation of preference, we can define the relation of incomparability in order to capture the inability of the decision-maker to choose.

Definition 3 (Relation of incomparability)**.** *Let* $\texttt{Alt}$ *be a set of alternatives and* $\succsim$ *a weak relation of preference over* $\texttt{Alt}$*. We define the relation of incomparability* $? \ \subseteq \ \texttt{Alt} \times \texttt{Alt}$ *such that:* $\forall(x,\ y) \in \texttt{Alt}^2,\ x\,?\,y \Leftrightarrow \neg(x \succsim y)$ *and* $\neg(y \succsim x)$*.*
If $x\ ?\ y$*, we say that "the alternatives* x *and* y *are incomparable".*

Two alternatives are incomparable if the decision-maker find difficult the comparison and he may decline to commit himself to a strong preference judgment while not being sure that he regards x and y as equally desirable (or undesirable).

The goal of the decision-maker is to select one of the best alternatives.

Definition 4 (Optimality)**.** *Let* $\texttt{Alt}$ *be a finite set of alternatives and* $\succsim$ *a weak relation of preference over* $\texttt{Alt}$*. An alternative* $x \in \texttt{Alt}$ *is* ***optimal over*** $\texttt{Alt}$ *iff:* $\forall y \in \texttt{Alt},\ x \succsim y$*.*

Remark 3. *If the relation of preference* $\succsim$ *is a total preorder, i.e.* $\forall x, y \in \texttt{Alt}^2,\ (x \succsim y) \vee (y \succsim x)$*, there is an underlying assumption such that all the alternatives are comparable and so, there is at least one optimal alternative over* $\texttt{Alt}$*.*

In the general case, when the preorder is partial, there is not always an optimal alternatives over a given set. The non-dominance notion, which is less restrictive, allows to distinguish the alternatives which are not defeated by other ones.

Definition 5 (Non-dominance)**.** *Let* $\texttt{Alt}$ *be a finite set of alternatives and* $\succsim$ *a weak relation of preference over* $\texttt{Alt}$*. An alternative* $x \in \texttt{Alt}$ *is* ***non-dominated over*** $\texttt{Alt}$ *iff:* $\forall y \in \texttt{Alt},\ \neg(y \succ x)$*.*

Remark 4. *The property of non-dominance is less restrictive that the property of optimality. Indeed, any optimal alternative over* $\texttt{Alt}$ *is non-dominated over* $\texttt{Alt}$ *but the reciprocal is not necessary true.*

Remark 5. *There is always at least one non-dominated alternative over a given set of alternatives.*

2.2 Choice

Contrary to the preferences of the decision-maker which are inaccessible, his decisions are observable. The choice function is the function used in the decision-making process in order to select a subset of alternatives.

Definition 6 (Choice function)**.** *Let* $\texttt{Alt}$ *be a finite set of alternatives. The* ***choice function*** $c : 2^{\texttt{Alt}} \rightarrow 2^{\texttt{Alt}} \cup \{\theta\}$ *is defined such that if* $B \subseteq E$ *then:*

1. $c(B) \subseteq (B \cup \{\theta\})$*;*
2. *and if* $B \neq \emptyset$ *then* $c(B) \neq \emptyset$*.*

Contrary to the classical definition, our choice function can return $\{\theta\}$ which represents the non-choice. Thus, we consider that the non-choice is a decision, i.e. the result of the behaviour of the decision-maker, and it is not an alternative ranked by the preferences.

The choice of the decision-maker is made in accordance with his preferences. Indeed, an optimal choice consists of selecting an alternative which is (weakly) preferred to the other ones.

Definition 7 (Optimal choice)**.** *Let* $\texttt{Alt}$ *be a finite set of alternatives,* $\succsim$ *a total relation of preference (cf. Section 2.1). The* ***optimal choice function*** c_{opt} *is a choice function defined on any set* $B \subseteq \texttt{Alt}$ *such that:* $c_{opt}(B) = \{x \in B \mid \forall y \in B,\ x \succsim y\}$*.*

The optimal choice function returns a subset of alternatives which are optimal (cf. Def. 4).

Remark 6. *An optimal choice function is a choice function (cf. Def. 6). Indeed, since the relation* $\succsim$ *is transitive and total, there exists at least one optimal alternative over* B *if* $B \neq \emptyset$ *and so,* $c_{opt}(B) \neq \emptyset$ *if* $B \neq \emptyset$*.*

If the underlying preferences are partial, then the notion of optimal choice is useless. We need to define the non-dominated choice function.

Let us consider a decision-maker dealing with a dilemma, i.e. $(\{x, y\},\ \succsim)$ such that $(x\ ?\ y)$. There is two ways to solve this dilemma. Either the decision-maker adopts a *laxist* attitude and $c(\{x, y\}) = \{x, y\}$. Or the decision-maker adopts a *rigorous* attitude and $c(\{x, y\}) = \{\theta\}$.

Definition 8 (Laxist non-dominated choice function)**.** *Let* $\texttt{Alt}$ *be a set of alternatives and* $\succsim$ *a preorder over the set of alternatives. The* ***laxist non-dominated choice function*** c_{NDL} *is choice function defined with* $B \subseteq \texttt{Alt}$ *such that:*

$$c_{NDL}(B) = \{\ x \in B \mid \nexists y \in B,\ y \succ x\ \}$$

Remark 7. *If the relation* $\succsim$ *is total,* $c_{NDL}(B) = c_{opt}(B)$*.*

If the decision-maker adopts a laxist non-dominated choice function, then an observer cannot infer the underlying relation of preference. Indeed, an observer cannot distinguish the indifference (cf. Def. 2) and the incomparability (cf. Def. 3) between these alternatives. Indeed,

$$c_{NDL}(\{x, y\}) = \{x, y\} \Rightarrow \quad (x \sim y) \text{ or } (x\ ?\ y)$$

Definition 9 (Rigorist non-dominated choice)**.** *Let* $\texttt{Alt}$ *be a set of alternatives and* $\succsim$ *a preorder over the set of alternatives. The* ***rigorous non-dominated choice function*** c_{NDR} *is choice function defined with* $B \subseteq \texttt{Alt}$ *such that:*

$$c_{NDR}(B) = \left\{ \begin{array}{l} c_{NDL}(B) \textit{ iff } \forall(x,\ y) \in (c_{NDL}(B))^2,\ x \sim y \\ \{\theta\} \textit{ else} \end{array} \right\}$$

If the decision-maker adopts a rigorous non-dominated choice function, then an observer can distinguish the indifference and the incomparability between two alternatives: $c_{NDR}(\{x, y\}) = \{x, y\} \Rightarrow x \sim y$ and $c_{NDR}(\{x, y\}) = \{\theta\} \Rightarrow (x\ ?\ y)$.

3 Multi-attribute decision

In a multiattribute decision problem, we make the assumption that the alternatives are defined in the same space, i.e. a set of attributes which is defined by a concept. In this way, the alternatives are instances of this concept. Firstly, we define the notions of concept and instance and some associated functions. Secondly, we define the lexicographic relation of preference.

3.1 Concept

A **concept** is a shape defined by a set of attributes.

Definition 10 (Concept)**.** *Let $(\mathtt{Att}_i)_{i \in I}$ be a family of sets indexed by integers in I. A **concept** C^I is the Cartesian product $C^I = \prod_{i \in I} \mathtt{Att}_i$.*

In a concept, each set, called attribute, is a domain of definition.

A **subconcept** is obviously a concept restricted to a subset of attributes.

Definition 11 (Subconcept)**.** *Let $(\mathtt{Att}_i)_{i \in I}$ be a family indexed by integers in I and $C^I = \prod_{i \in I} \mathtt{Att}_i$ is the corresponding concept. Let $J \subset I$, a **subconcept** of C^I **over** J is the Cartesian product C^J defined such that: $C^J = \prod_{j \in J} \mathtt{Att}_j$.*

3.2 Instance

An **instance** is a concrete object typed by a concept (cf. Def. 10). An instance has values within the attributes defined by the associated concept.

Definition 12 (Instance)**.** *Let $C^I = \prod_{i \in I} \mathtt{Att}_i$ be a concept. An instance $x_{C^I} \in C^I$ is a vector of values $(v_1, \ldots, v_n)$ where $\forall i \in I$, $v_i \in \mathtt{Att}_i$.*

We consider here that the attributes are the domains of definition for the values. In this way, the alternatives are instances of a concept. The set of alternatives $\mathtt{Alt}_{C^I}$ is composed of instances of the concept C^I. This a subset of the Cartesian product $\mathtt{Alt}_{C^I} \subseteq C^I$. It is worth noticing that there is not necessary an instance for all the possible values.

3.3 Projection and selection

In order to define multiattribute preferences, we first introduce some functions to manipulate instances and set of instances.

In the following definitions, we consider a concept C^I and a subconcept C^J with $J \subset I$ (cf. Def. 10 and 11).

The projection of x over the attributes indexed by integers in I is an element of the set C^J with the same instanciation as x for all the attributes indexed by integers in J.

Definition 13 (Projection)**.** *Let $x_{C^I} = (v_{i_1}, \ldots, v_{i_n}) \in C^I$ be an instance where $i_k \in I$. A **projection** of x_{C^I} over J with $J \subseteq I$ is the function $\pi_J : C^I \to C^J$ defined such that: $\pi_J(x_{C^I}) = (v_{j_1}, \ldots, v_{j_m}) \in C^J$ where $j_k \in J$*

According to this definition, the projection of an instance over a set of attributes consists of the values of this instance for this set of attributes without considering the other attributes.

As we have defined the projection of an instance, we can also define here the projection of a set of instances.

Definition 14 (Projection of a set)**.** *Let $\mathtt{Alt}_{C^I}$ be a set of alternatives with $\mathtt{Alt}_{C^I} \subseteq C^I$. $P_J : \mathcal{P}(C^I) \to \mathcal{P}(C^J)$ is the **projection of the set** $\mathtt{Alt}_{C^I}$ defined such that: $P_J(\mathtt{Alt}_{C^I}) = \{\pi_J(x) \mid x \in \mathtt{Alt}_{C^I}\}$.*

The projection of a set of instances is the set of the projections for all these instances.

We define the selection of a set of alternatives $\mathtt{Alt}_{C^I}$ with respect to an instance x_{C^J} of C^J as the set of elements of $\mathtt{Alt}_{C^I}$ which are projected on J and which are equals to x_{C^J}.

Definition 15 (Selection with respect to an instance)**.** *Let $\mathtt{Alt}_{C^I}$ be a set of alternatives with $\mathtt{Alt}_{C^I} \subseteq C^I$ and $x_{C^J} \in C^J$ be an instance of C^J with $J \subseteq I$. A **selection** of $\mathtt{Alt}_{C^I}$ with respect to x_{C^J} is the function $\sigma_{x_{C^J}} : \mathcal{P}(\mathtt{Alt}_{C^I}) \to \mathcal{P}(\mathtt{Alt}_{C^I})$ defined such that: $\sigma_{x_{C^J}}(\mathtt{Alt}_{C^I}) = \{x_{C^I} \in \mathtt{Alt}_{C^I} \mid \pi_J(x_{C^I}) = x_{C^J}\}$*

In the same way we have defined the selection with respect to an instance, we define the selection with respect to a set of instances.

Definition 16 (Selection with respect to a set of instances)**.** *Let $\mathtt{Alt}_{C^I}$ be a set of alternatives with $\mathtt{Alt}_{C^I} \subseteq C^I$ and $\mathtt{Alt}_{C^J}$ be a set of subalternatives with $\mathtt{Alt}_{C^J} \subseteq C^J$. The **selection** of $\mathtt{Alt}_{C^J}$ **with respect to** $\mathtt{Alt}_{C^I}$ is the function $\sigma_{\mathtt{Alt}_{C^J}} : \mathcal{P}(\mathtt{Alt}_{C^I}) \to \mathcal{P}(\mathtt{Alt}_{C^I})$ defined such that: $\sigma_{\mathtt{Alt}_{C^J}}(\mathtt{Alt}_{C^I}) = \bigcup_{x_{C^J} \in \mathtt{Alt}_{C^J}} \sigma_{x_{C^J}}(\mathtt{Alt}_{C^I})$*

The selection of a set of alternatives wit respect to a set of instances is the union of the selections.

3.4 Multi-attribute preferences

In order to define the preferences over multi-attribute alternatives, we consider the preference relation on attributes capturing their relative importance and a family of preference relations indexed on each attribute capturing the penchant with respect to their values.

Preferences over multi-attribute alternatives can easily be defined from preferences over attributes and preferences over values.

Definition 17 (Lexicographic preferences)**.**

Let $C = \prod_{i \in I} \mathtt{Att}_i$ be a concept, $\succsim_I$ the preference relation over attributes and $(\succsim_{\mathtt{Att}_i})_{i \in I}$ the family of preference relations on each attribute. The multi-attribute preference relation over instances of C is the relation $\succsim_C$ defined such as for each pair of alternatives $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$:

- $x \succ_C y \Leftrightarrow \exists l \in I,\ x \succ_{\mathtt{Att}_l} y$ *and* $\forall k \in I \mid \neg(\mathtt{Att}_l \succ_I \mathtt{Att}_k) \Rightarrow (x \succsim_{\mathtt{Att}_k} y)$
- $x \sim_C y \Leftrightarrow \forall k \in I, y \sim_{\mathtt{Att}_k} x$

Remark 8. *If the preference relation over the set of attributes $\succsim_I$ is strong and complete and if the preference relations $(\succsim_{\mathtt{Att}_i})_{i \in I}$ are complete then the preference relation over instances $\succsim_C$ is complete.*

Remark 9. *If the preference relation over the set of attributes $\succsim_I$ is weak or partial then the preference relation over instances $\succsim_C$ is not necessarily complete.*

4 Dilemma

A dilemma occurs when the preference relation of the decision-maker is partial. If non-dominated alternatives cannot be compared, then the rigorous choice function will return a non-choice.

We claim that theses dilemmas have a minor impact in most of the practical situations. We discuss here this assumption.

4.1 Protocol

In the following experiments, we consider a concept $\mathtt{C}$ with 10 attributes having binary values. For each experiments, we randomize:

- a partial preference relation over the attributes (see line 0 in Tab. 4). We control the incomparability ratio of the relation. Moreover, the preference relations over the values are total and independent;
- a set of alternatives for which we control the filling ratio.

For each experiment, we look if the choice $\mathtt{c}_{NDR}(\mathtt{Alt})$ returns an alternative or a non-choice (i.e. θ).

In Fig. 1, we have tested different incomparability ratio (between 0 and 1) and different filling ratio for the set of alternatives (between 0.2 and 1). For each couple of ratio (a point in our graph) we run 1000 experiments and so, the total number of decision-making problems we consider is 50 000.

4.2 Results

In Fig. 1, we observe that non-choice arises when few attributes are comparable and when the number of available alternatives is low. Indeed, it is likely to find an optimal alternative when they are a lot of alternatives and/or they are not too much incomparabilities. Even if the filling ratio is low, the non-choice rarely arises (in less than 20 % of the cases) if the ratio of incomparability is not too high (less than 0.65). In conclusion, most of the dilemmas can be solved by considering concrete alternatives.

5 Multi-attribute decision aid game

We model here the decision aiding process by a dialogue where an analyst asks questions to a decision maker in order to collect and deduce his preferences. The analyst suggests attributes and values which can be chosen by the decision maker based upon his preferences. The analyst aims at finding a preferred alternative.

5.1 Oracles

In our multi-attribute decision aid game, we consider two oracles :

- $\mathtt{c}_1$ give the subset of the most important attributes in a set of proposals ;
- $\mathtt{c}_2$ give the best instances of a concept (or subconcept) in a set of proposals.

These oracles are choice functions as defined in section 2.2.

5.2 Multi-attribute decision aid algorithm

We now present our algorithm for multi-attribute decision aid based on an active learning mechanism of lexicographic preferences. In our approach, the analyst wants to infer a sufficient part of the preferences in order to find an alternative which satisfy his choice function. For this purpose, the analyst queries the decision-maker with the help of two oracles (See Section 5.1).

The algorithm 1 uses as inputs :

- $C = \prod_{i \in I} \mathtt{Att}_i$, the concept;
- $\mathcal{A} = \{\mathtt{Att}_i\}_{i \in I}$, the set of attributes of this concept ;
- and $\mathtt{Alt}_C$, the set of alternatives (i.e. available instances) defined in C.

The loop variables are :

- $\mathtt{Alt}^{tmp}$, the remaining alternatives (not excluded) ;
- $\mathcal{A}^{tmp}$, the attributes we did not already consider and on which the remaining alternatives ($\mathtt{Alt}^{tmp}$) have at learning two different values;
- $\mathcal{A}^{max}$, the set of predominant attributes in $\mathcal{A}^{tmp}$;
- $\mathtt{Alt}^{max}$, the remaining alternatives on $\mathcal{A}^{max}$.

Additionally we consider two oracles (See section 5.1):

- $\mathtt{c}_1$ (line 4 of the algorithm) selects the predominant attributes ($\mathcal{A}^{max}$) in the remaining attributes ($\mathcal{A}^{tmp}$).
- $\mathtt{c}_2$ (line 6 of the algorithm) selects the preferred tuples in the remaining alternatives ($\mathtt{Alt}^{tmp}$) projected on the set of predominant attributes $\mathcal{A}^{max}$ (cf. Def. 14).

Algorithm 1: Algorithm for active learning mechanism of lexicographic preferences

```
   Data: $\mathcal{A} = \{\mathtt{Att}_i\}_{i \in I}$, $C = \prod_{i \in I} \mathtt{Att}_i$, $\mathtt{Alt}_C \subseteq C$
   Result: $\mathtt{Alt}' \subseteq \mathtt{Alt}_C$, preferred alternatives
1  $\mathtt{Alt}^{tmp} \leftarrow \mathtt{Alt}_C$ ;
2  $\mathcal{A}^{tmp} \leftarrow \{\mathtt{Att}_i \in \mathcal{A} \mid card(\pi_{\{i\}}(\mathtt{Alt}^{tmp})) > 1\}$ ;
3  while $(\mathcal{A}^{tmp} \neq \emptyset) \wedge (|Alt^{tmp}| \neq 1)$ do
4      $\mathcal{A}^{max} \leftarrow \mathtt{c}_1(\mathcal{A}^{tmp})$ ;
5      // NB : we suppose $\mathcal{A}^{max} \neq \{\theta\}$;
6      $\mathtt{Alt}^{max} \leftarrow \mathtt{c}_2(P_{\mathcal{A}^{max}}(\mathtt{Alt}^{tmp}))$ ;
7      if $Alt^{max} = \{\theta\}$ then
8          return $\{\theta\}$
9      end
10     $\mathtt{Alt}^{tmp} \leftarrow \sigma_{\mathtt{Alt}^{max}}(\mathtt{Alt}^{tmp})$ ;
11     // NB : set of ex-aequo on $\mathcal{A}_{max}$;
12     $\mathcal{A}^{tmp} \leftarrow \mathcal{A}^{tmp} - \mathcal{A}^{max}$ ;
13     $\mathcal{A}^{tmp} \leftarrow \{\mathtt{Att}_i \in \mathcal{A}^{tmp} \mid card(\pi_{\{i\}}(\mathtt{Alt}^{tmp})) > 1\}$;
14 end
15 return $Alt^{tmp}$
```

In the worst case, the algorithm needs $|\mathcal{A}|$ queries to $\mathtt{c}_1$ and $|\mathcal{A}|$ queries to $\mathtt{c}_2$.

5.3 Example

In this section, we illustrate our algorithm with a toy example. Let us consider a decision maker who wants to buy a car. We define the concept $\mathtt{Car}$ with the attributes described in Table 1.

Attribute	Values
Brand	{Peugeot, Opel, Ford}
Price	{Cheap, Moderate, Expensive}
Availability	{Immediately, Later}
Motorization	{Diesel,Petrol}

Table 1. Definition of the attributes of $\mathtt{Car}$

The concept $\mathtt{Car}$ is the cartesian product of these attributes, i.e. $Car = Brand \times Price \times Availability \times Motorization$. These attributes are modelled with discrete values, in particular prices.

The alternatives for this problem, available instances of the concept $\mathtt{Car}$, are listed in Table 2.

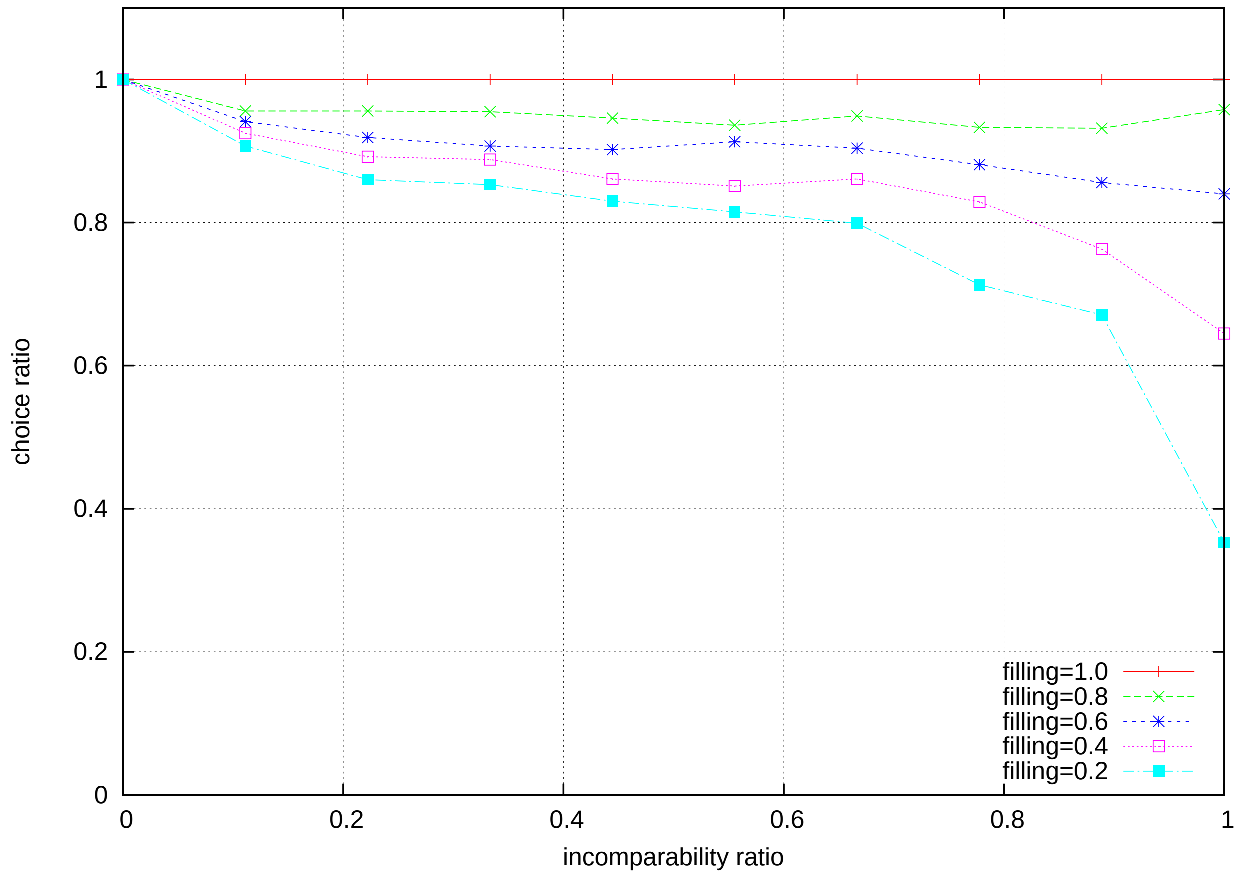


Figure 1. Choice ratio of $\mathtt{c}_{NDL}(\mathtt{Alt})$ in function of incomparability ratio of $\succ_I$ for different filling ratio of Alt

Id	Brand	Price	Availability	Motorization
a_1	Peugeot	Expensive	Immediately	Diesel
a_2	Peugeot	Cheap	Immediately	Petrol
a_3	Ford	Expensive	Later	Diesel
a_4	Ford	Moderate	Immediately	Petrol
a_5	Opel	Expensive	Immediately	Diesel
a_6	Opel	Moderate	Later	Diesel
a_7	Opel	Moderate	Later	Petrol

Table 2. The alternatives for the decision problem

We finish to describe our decision problem in considering the decision maker having the preferences listed in Table 3.

Attribute set	$Brand \succ Price$ $Brand \succ Motorization$ $Price \succ Availability$ $Motorization \succ Availability$
Brand	$Peugeot \sim Opel$ $Opel \succ Ford$ $Peugeot \succ Ford$
Price	$Cheap \succ Moderate$ $Moderate \succ Expensive$
$Availability$	$Immediately \succ Later$
$Motorization$	$Petrol \succ Diesel$

Table 3. The preferences of the decision maker

In order to identify the preferable alternative(s), we follow the algorithm 1. In Fig. 2, we focus on the trace of oracle calls and remaining alternatives during the dialogue.

1. $\mathtt{c}_1(\{Brand, Price, Availability, Motorization\})$
   $= \{Brand\}$

2. $\mathtt{c}_2(\{Peugeot, Opel, Ford\})$
   $= \{Peugeot, Opel\}$
   $\mathtt{Alt}^{tmp} = \{a_1, a_2, a_5, a_6, a_7\}$

3. $\mathtt{c}_1(\{Price, Availability, Motorization\})$
   $= \{Price, Motorization\}$

4. $\mathtt{c}_2(\{(Expensive, Diesel),\ (Cheap, Petrol),$
   $(Moderate, Diesel), (Moderate, Petrol)\})$
   $= \{(Moderate, Diesel), (Cheap,)\}$
   $\mathtt{Alt}^{tmp} = \{a_2, a_6\}$

5. $\mathtt{c}_2(\{Immediately, Later\})$
   $= \{Immediately\}$
   $\mathtt{Alt}^{tmp} = \{a_2\}$

Figure 2. Oracle calls

We can consider that the decision aiding dialogue starts with a query about which attribute(s) is/are dominant. The decision maker answers that the brand of the car is crucial for her (call #1) and she prefers either a $Peugeot$ or a $Opel$ (call #2). The alternatives a_3, a_4 are then dominated and removed. Then we ask the decision maker what is the next most important attribute(s). He answers that price and motorization are both more important than the availability (call #3). So, we propose the four available pair (cf. call #4). The decision maker prefers pairs $(Moderate, Diesel)$ and $(Cheap, Petrol)$ but can not choose. Since she is laxist, we ask her a last question about her preferences about the only remaining attribute (call #5) and finally the remaining alternative is a_2, which is the only non-dominated instance considering the decision maker preferences (cf. Tab. 3).

5.4 Instantiations

In this section, we study different relations of preference and different choice functions for the multi-attribute decision aid game.

We consider here strong relation on attributes (cf. Tab. 4). In other words, we consider incomparability between attributes but we do not envisage indifference.

- In the case #1, we consider strong and complete preferences over the set of attributes and on each attribute. Therefore, the preferences over the set of instances is strict and total (cf. Section 3.4). The decision-maker uses an optimal choice function for the attributes and the instances. Therefore, the preference relation over the instances is strong and complete, and there is a single optimal alternative.
- The case #2 is quite similar. The preferences over the set of attributes are no necessary strong but they are complete. In the general case, preferences over instances are large and so, there may be several optimal alternatives.
- In the case #3, we consider weak partial preferences on each attribute. It means that some alternatives can be incomparable. The optimal choice function is no suitable in this case so we assume the decision-maker uses a laxist non-dominated choice function (cf. Def. 8). Therefore, the preference relation over instances is partial, and so there always is at least one non-dominated alternative.
- The case #4 make the same assumptions as case #3 on the preferences and the decision-maker uses a rigorous non-dominated choice function (cf. Def. 9). There can be zero, one or several acceptable alternatives.
- In the case #5, we loss the completeness of the relation over the set of attributes and we consider weak and complete preferences over each attribute. Preferences over alternatives can be weak and partial as in case #4. There always is at least one non-dominated alternative as in case #3.
- The case #6 differs on #5 by the nature of the choice function which is here rigorous. That is the reason why it is possible that some alternatives are acceptable for the decision-maker.

6 Related Works

Preference learning has received increasing attention in Artificial Intelligence in recent years. Fürnkranz and Hüllermeier propose in [5] a typology which distinguishes label ranking, instance ranking and object ranking. Our work is concerned by object ranking in an active learning mode based upon a dialogue between a decision maker and an analyst [7] rather than a given data set.

Among the works on learning ordinal preferences in multi-attribute domains, Chevaleyre *et al.* [1] consider conditional preference networks (CP-nets). They study the learning of CP-nets in a

#	$\succ_I$	$\mathtt{c}_1$	$(\succsim_{\mathtt{Att}_i})_{i\in I}$	$\succsim_C$	$\mathtt{c}_2$	Solutions
0 (cf. section 4)	strong partial		strong complete	lexicographic strong partial		$1 \geq \lvert S\rvert \geq 0$
1	strong complete	$\mathtt{c}_{opt}$	strong complete	lexicographic strict total	$\mathtt{c}_{opt}$	$\lvert S\rvert = 1$
2	strong complete	$\mathtt{c}_{opt}$	weak complete	lexicographic weak total	$\mathtt{c}_{opt}$	$\lvert S\rvert \geq 1$
3	strong complete	$\mathtt{c}_{opt}$	weak partial	lexicographic weak partial	$\mathtt{c}_{NDL}$	$\lvert S\rvert \geq 1$
4	strong complete	$\mathtt{c}_{opt}$	weak partial	lexicographic weak partial	$\mathtt{c}_{NDR}$	$\lvert S\rvert \geq 0$
5	strong partial	$\mathtt{c}_{NDL}$	weak complete	lexicographic weak partial	$\mathtt{c}_{NDL}$	$\lvert S\rvert \geq 1$
6	strong partial	$\mathtt{c}_{NDL}$	weak complete	lexicographic weak partial	$\mathtt{c}_{NDR}$	$\lvert S\rvert \geq 0$

Table 4. Instantiations of multi-attribute decision aid game

	Learning mode	Preferences	Input
Dombi *et al.* [4]	passive and active	total lexicographic order	preferences over attributes
Chevaleyre *et al.* [1]	passive and active	CP-nets	preferences over instances
Yaman *et al.* [10]	passive	total lexicographic order	preferences over instances
Delecroix *et al.*	active	lexicographic order (total or partial)	preferences over attributes and (sub-)instances

Table 5. Related works

passive mode and in an active mode. Contrary to our work, the CP-net can translate the dependency relationships between attributes but it cannot express the relative importance of attributes.

Dombi *et al.*[4] deal with learning preferences in a passive mode and in an active mode in order to infer a lexicographic order on attributes. In our case, we focus on finding one preferred alternative. The alternatives are not input data of the problem in [4]. Our algorithm depends on these alternatives. Moreover, we do not restrict ourselves to a total lexicographic order. However, we envisage a partial lexicographic order which allows us to model dilemmas. Finally, we do not make any assumption about the preferences over the attribute values.

In their work, Yaman *et al.* [10] address the problem of learning lexicographic preferences in a multiattribute domain based on a democratic approximation. They propose two methods:

- *variable voting* consists of deducing the ranking of attributes based on a series of observations. Then, the algorithm deduces the preferences over alternatives with the help of a voting mechanism.
- *model voting* is based on a Bayesian approach. After having identify and weight a set of lexicographic preference orders which are consistent with the observations, a vote is performed in order to deduce the preferences over alternatives.

These methods use passive learning algorithms.

As stated in Tab. 5, we distinguish our work on two issues. On one hand, we take into account the dilemmas and so, we make less assumptions on the preferences. On the other hand, we distinguish ourselves due to the interaction with the decision-maker. We ask questions about the relative importance of attributes and we ask questions about preferences over tuples of values, and so we avoid to ask questions on complex instances.

7 Conclusion

In this paper, we have modelled the decision aiding process by a dialogue where an analyst asks questions to a decision maker in order to collect and deduce his preferences. The analyst suggests attributes and values which can be chosen by the decision maker based upon his preferences. The analyst aims at finding a preferred alternative. For this purpose, we make the assumption that the preferences of the decision-maker can be captured by a lexicographic partial order. Even if the incomparabilities between attributes can lead to dilemmas, most of the choices between the values of these attributes allow to solve these situations. Based on this idea, we have proposed an algorithm for active learning of (partial) lexicographic preferences.

In future work, we aims at evaluating our adaptive algorithm in a theoretical and experimental way. Our practical objective is to identify a preferred alternative by asking few simple questions. For this purpose, we need to introduce a metric for evaluating the size of questions as done in [1].

REFERENCES

[1] Yann Chevaleyre, Frédéric Koriche, Jérôme Lang, Jérôme Mengin, and Bruno Zanuttini. Learning ordinal preferences on multiattribute domains: the case of cp-nets, 2010.

[2] Fabien Delecroix, Maxime Morge, and Jean-Christophe Routier, *Agreement Technologies*, chapter A Virtual Selling Agent which is Persuasive and Adaptive, 621–635, Springer Verlag, 2012.

[3] Fabien Delecroix, Maxime Morge, and Jean-Christophe Routier, 'A virtual selling agent which is proactive and adaptive', in *Proc. of of the 10th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 12)*, Advances in Intelligent and Soft-Computing, pp. 57–66, Salamanca, (April 2012). Springer.

[4] József Dombi, Csanád Imreh, and Nándor Vincze, 'Learning lexicographic orders', *European Journal of Operational Research*, **183**(2), 748 – 756, (2007).

[5] Johannes Fürnkranz and Eyke Hüllermeier, 'Preference learning: An introduction', in *Preference Learning*. Springer-Verlag.

[6] Miquel Montaner, Beatriz López, and Josep Lluís De La Rosa, 'A taxonomy of recommender agents on the Internet', *Artif. Intell. Rev.*, **19**, 285–330, (June 2003).

[7] Wassila Ouerdane, Nicolas Maudet, and Alexis Tsoukiàs, 'Argumentation theory and decision aiding', *International Series in Operations Research and Management Science*, **142**, 177–208, (2010).

[8] Paul Resnick and Hal R. Varian, 'Recommender systems', *Commun. ACM*, **40**(3), 56–58, (March 1997).

[9] Burr Settles, 'Active learning literature survey', Computer Sciences Technical Report 1648, University of Wisconsin–Madison, (2009).

[10] Fusun Yaman, Thomas J. Walsh, Michael L. Littman, and Marie desJardins, 'Democratic approximation of lexicographic preference models', *Artif. Intell.*, 1290–1307, (2011).