DA2PL'2014: from Multiple Criteria Decision Aid to Preference Learning

Vincent Mousseau, Marc Pirlot

CENTRALE
PARIS

UMONS
Université de Mons

DA2PL'2014

from Multiple Criteria Decision Aid to Preference Learning

Ecole Centrale Paris
November 20-21, 2014

edited by Vincent Mousseau and Marc Pirlot

Table of Content

Welcome

Vincent Mousseau (Ecole Centrale Paris) and Marc Pirlot (UMONS) are welcoming you to the second DA2PL Workshop. The first edition took place in Mons (Belgium) in November 2012. The aims of this serie of 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 Ecole Centrale Paris.

Vincent Mousseau and Marc Pirlot

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.

The theme of the DA2PL 2014 workshop is (specifically but not excusively) devoted to "*preference models with interacting criteria*".

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 10 sessions including:

- 6 invited lectures of internationally recognized scholars,
- 12 refereed research presentations,
- a poster session.

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 workshop is also supported by the French GDR RO (CNRS) - Pôle : Décision : Modélisation, Prévision, Evaluation (DMPE). The support of Ecole Centrale Paris (Direction de la Recherche) is also gratefully acknowledged.

Organization

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

Program committee

- Raymond Bisdorff (University of Luxembourg, Luxembourg),
- Craig Boutillier (University of Toronto, Canada),
- Denis Bouyssou (Paris Dauphine University, France),
- Robert Busa-Fekete (Marburg University, Germany),
- Olivier Cailloux (University of Amsterdam, Netherlands),
- Yann Chevaleyre (University of Paris North, France),
- Yves Crama (University of Liege, Belgium),
- Bernard De Baets (Ghent University, Belgium),
- Yves De Smet (Université libre de Bruxelles, Belgium),
- Krzysztof Dembczyn'ski, Poznan University of Technology, Poland,
- Luis Dias (University of Coimbra, Portugal),
- Philippe Fortemps (University of Mons, Belgium),
- Michel Grabisch (University Paris 1, France),
- Salvatore Greco (University of Catania, Italy),
- Eyke Hullermeier (Marburg University, Germany),
- Christophe Labreuche (Thales, France),
- Patrick Meyer (Telecom Bretagne, France),
- Vincent Mousseau (Ecole Centrale, Paris),
- Patrice Perny (Pierre and Marie Curie University, France),
- Marc Pirlot (University of Mons, Belgium),
- Fred Roberts (DIMACS, Rutger University, USA),
- Ahti Salo (Aalto University, Finland),
- Roman Slowinski (Poznan University of Technology, Poland),
- Alexis Tsoukias (Paris Dauphine University, France),
- Aida Valls (Universitat Rovira I Virgili, Catalonia, Spain),
- Paolo Viappiani (Pierre and Marie Curie University, France)

Organizing committee

- Olivier Cailloux, University of Amsterdam, Netherlands
- Sylvie Guillemain, Laboratoire de Génie Industriel, Ecole Centrale Paris, France
- Jinyan Liu, Laboratoire de Génie Industriel, Ecole Centrale Paris, France
- Delphine Martin, Laboratoire de Génie Industriel, Ecole Centrale Paris, France
- Vincent Mousseau, Laboratoire de Génie Industriel, Ecole Centrale Paris, France
- Corinne Ollivier, 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
- Olivier Sobrie, Laboratoire de Génie Industriel, Ecole Centrale Paris, France and Université de Mons, Belgium

List of participants

- Jamal Atif, Lamsade, Université Paris Dauphine, jamal.atif@dauphine.fr
- Alexandre Aubry, Place des Leads, alexandre.aubry@placedesleads.com
- Nawal Benabbou, LIP6, Université Pierre et Marie Curie, benabbou@poleia.lip6.fr
- Raymond Bisdorff, Université du Luxembourg, raymond.bisdorff@uni.lu
- Jean-Claude Bocquet, LGI, Ecole Centrale Paris, jean-claude.bocquet@ecp.fr
- Endre Boros, RUTCOR, Rutgers University, endre.boros@gmail.com
- Craig Boutillier, University of Toronto, cebly@cs.toronto.edu
- Denis Bouyssou, Lamsade, Université Paris Dauphine, bouyssou@lamsade.dauphine.fr
- Valérie Brison, MATHRO, UMONS, Valerie.Brison@umons.ac.be
- Olivier Cailloux Université de technologie de Compiègne, olivier.cailloux@uva.nl
- Yves Crama, Université de Liège, Yves.Crama@ulg.ac.be
- Krzysztof Dembczynski, Poznan University of Technology, Krzysztof.Dembczynski@cs.put.poznan.pl
- Sébastien Destercke, Université de technologie de Compiègne, sebastien.destercke@hds.utc.fr
- Mahdi Fathi, LGI, Ecole Centrale Paris, mahdi.fathi@ecp.fr
- Hugo Gilbert, LIP6, Université Pierre et Marie Curie, gilbert@poleia.lip6.fr
- Bénédicte Goujon Thales Research & Technology France, benedicte.goujon@thalesgroup.com
- Michel Grabisch, Université Paris 1, michel.grabisch@univ-paris1.fr
- Alain Guénoche, Institut de Mathématiques de Marseille (I2M - CNRS), alain.guenoche@univ-amu.fr
- Sylvie Guillemain, LGI, Ecole Centrale Paris, sylvie.guillemain@ecp.fr
- Massimo Gurrieri, MATHRO, UMONS, Massimo.Gurrieri@umons.ac.be
- Allel Hadjali, LIAS/ENSMA, Poitier, allel.hadjali@ensma.fr
- Céline Hudelot, MAS, Ecole Centrale Paris, celine.hudelot@ecp.fr
- Eyke Hüllermeier, University of Paderborn, eyke@Mathematik.Uni-Marburg.de
- Marija Jankovic, LGI, Ecole Centrale Paris, marija.jankovic@ecp.fr
- Fabien Labernia, Lamsade, Université Paris Dauphine, fabien.labernia@gmail.com
- Christophe Labreuche, Thales Research & Technology France, christophe.labreuche@thalesgroup.com
- Jérôme Lang, Lamsade, Université Paris Dauphine, lang@lamsade.dauphine.fr
- Jinyan Liu, LGI, Ecole Centrale Paris, jinyan.liu@ecp.fr
- Delphine Martin, LGI, Ecole Centrale Paris, delphine.martin@ecp.fr
- Brice Mayag, Lamsade, Université Paris Dauphine, brice.mayag@ecp.fr
- Jérôme Mengin, IRIT, Jerome.Mengin@irit.fr
- Patrick Meyer, Telecom Bretagne, patrick.meyer@telecom-bretagne.eu
- Tatyana Mironova, Telecom Bretagne, tatyana.mironova@telecom-bretagne.eu
- Vincent Mousseau, LGI, Ecole Centrale Paris, vincent.mousseau@ecp.fr
- Selin Ozpeynirci, Izmir University, Turkey, selin.ozpeynirci@ieu.edu.tr
- Ozgur Ozpeynirci, Izmir University, Turkey, ozgur.ozpeynirci@ieu.edu.tr

- Meltem Öztürk, Lamsade, Université Paris Dauphine, Meltem.Ozturk@dauphine.fr
- Patrice Perny LIP6, Université Pierre et Marie Curie, patrice.perny@lip6.fr
- Marc Pirlot, MATHRO, UMONS, Marc.Pirlot@umons.ac.be
- Elisabeth Rodríguez Heck, HEC Management School, University of Liège, elisabeth.rodriguezheck@ulg.ac.be
- Antoine Rolland, Université Lyon 2, arolland@eric.univ-lyon2.fr
- Olivier Sobrie, UMONS and LGI, Ecole Centrale Paris, olivier.sobrie@gmail.com
- Xavier Siebert, MATHRO, UMONS, xavier.siebert@umons.ac.be
- Bruno Teheux, Université du Luxembourg, bruno.teheux@uni.lu
- Mikhail Timonin, Queen Mary College, mikhail.timonin@gmail.com
- Alexis Tsoukiàs, Lamsade, Université Paris Dauphine, tsoukias@lamsade.dauphine.fr
- Dimitri Van Assche, Uinversité Libre de Bruxelles, dvassche@ulb.ac.be
- Paolo Viapianni, LIP6, Université Pierre et Marie Curie, Paolo.Viappiani@lip6.fr
- Angelina Vidali, LIP6, Université Pierre et Marie Curie, angvid@gmail.com
- Tairan Wang, LGI, Ecole Centrale Paris, tairan.wang@ecp.fr
- Paul Weng, LIP6, Université Pierre et Marie Curie, paul.weng@lip6.fr
- Michel Zam, KarmikSoft Research, m.zam@karmicsoft.com

CENTRALE PARIS

UMONS Université de Mons

DA2PL'2014
from Multiple Criteria Decision Aid to Preference Learning

Program overview

Thursday November 20th, 2014

9h30 Session 1

- Invited speaker: "*Preference Learning: Machine Learning meets MCDA*"
  Eyke Hüllermeier, Department of Computer Science, Universität Paderborn, 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, which are 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 provide a brief introduction to the field of preference learning and, moreover, to elaborate on its connection to multiple criteria decision aid.

10h30 Session 2

- "*On the use of copulas to simulate multicriteria data*",
  Jairo Cugliari, Antoine Rolland, Thi-Min-Tuy Tran, Lab. ERIC, Université Lyon 2

  Several methods have been proposed in the past decades to deal with Multicriteria Decision Aiding (MCDA) problems. However, a comparison between these methods is always arduous as the number of dataset proposed in the literature is very low. One of the limitations of the existing datasets is that generally MCDA method are dealing with very small sets of data; typically, a MCDA problem deals with a number of alternatives that does not exceed 20 or 30 and often less. Therefore, it should be interesting to propose a way to simulate new data based on some existing dataset, i.e. taking into account the potential links that should exist between the criteria. We introduce in this paper the use of the statistical functions named copula to simulate such data. A practical way to use copula is proposed, and the quality of the obtained data is discussed.

- "*Data Generation Techniques for Label Ranking*",
  Massimo Gurrieri, Philippe Fortemps, Xavier Siebert, Marc Pirlot, Nabil Aït-Taleb
  MATHRO, Faculté Polytechnique, UMONS
  In light of the lack of benchmark data for label ranking, experimentations are typically performed on data sets derived from classification or regression data sets. The generation of artificial datasets is however not trivial since instances have to be associated with rankings over a finite set of labels and attributes (i.e. the feature vector) have to be linked (correlated) with such rankings. This paper discusses and proposes datasets generation techniques in order to provide artificial datasets suitable for label ranking.

11h30 Coffee Break

12h00 Session 3

- Invited speaker: “*Boolean functions for classification: logical analysis of data*”,
  Yves Crama, University of Liège, Belgium
  Boolean functions are among the simplest and most fundamental objects investigated in mathematics. In spite, or because of their simplicity, they find applications in many scientific fields, including logic, combinatorics, operations research, artificial intelligence, computer science, game theory, engineering, and so forth. In this talk, we present a collection of Boolean models that have been developed over the last 25 years under the name of "Logical Analysis of Data" (or LAD) in order to handle a large variety of classification problems. We focus on the frequent situation where a decision-maker has observed a number of data points (say, vectors of binary attributes) which have been classified either as "positive" or as "negative" examples of a phenomenon under study. The task of the decision-maker is then to develop a classification system that allows her to assign one of the "positive" or "negative" qualifiers to any point that may be presented to her in the future, in a way that remains consistent with the initial observations. We first recall useful facts about partially defined Boolean functions and their extensions, and we introduce the main concepts and definitions used in the LAD framework: support (or "sufficient") sets of attributes, patterns (or "elementary classification rules"), theories (obtained by combining patterns), etc. We show how these building blocks can be used to develop simple interpretable classifiers that perform and generalize well in a variety of experimental situations. Moreover, we argue that these classifiers satisfy some minimal requirements for “justifiability". Finally, we clarify the relation between the LAD classifiers and certain popular classifiers used in the machine learning literature, such as those computed by nearest neighbor classification algorithms or decision trees.

13h00 Lunch

14h20 Group Photo session

14h30 Session 4

- Invited speaker: “*Learning and identifying monotone boolean functions*”,
  Endre Boros, Rutgers University, NJ, USA
  Numerous applications require the task of learning and/or identifying a hidden monotone Boolean function.In this talk, first we review several learning models and clarify the the corresponding learning complexity when the hidden function is known to be monotone. The considered models include extending a given partially defined Boolean function or one with missing bits within a specified class of monotone Boolean functions, and learning a certain type of monotone function using membership queries. In the second part of the talk we consider identification problems, which is a special case/extension (depending how one views it) of learning by membership queries. Identification of a monotone function means that we try to generate all of its minimal true (resp. maximal false) points. This problem is strongly related to Boolean dualization or equivalently to finding all minimal transversals of a hypergraph. In this talk we survey some of the related results, and provide a sample of the standard algorithmic techniques.

15h30 Coffee Break

16h00 Session 5

- "*Learning the parameters of a majority rule sorting model taking attribute interactions into account*", Olivier Sobrie[1,2], Vincent Mousseau[1] and Marc Pirlot[2]
  [1] LGI, Ecole Centrale Paris,
  [2] MATHRO, Faculté Polytechnique, UMONS
  We consider a multicriteria sorting procedure based on a majority rule, called MR-Sort. This procedure allows to sort each object of a set, evaluated on multiple criteria, in a category selected among a set of pre-defined and ordered categories. With MR-Sort, the ordered categories are separated by profiles which are vectors of performances on the different attributes. An object is assigned in a category if it is as good as the category lower profile and not better than the category upper profile. To determine if an object is as good as a profile, the weights of the criteria on which the object performances are better than the profile performances are summed up and compared to a threshold. In view of improving the expressiveness of the model, we modify it by introducing capacities to quantify the power of the coalitions. In the paper we describe a mixed integer program and a metaheuristic that give the possibility to learn the parameters of this model from examples of assignment. We test the metaheuristic on real datasets.

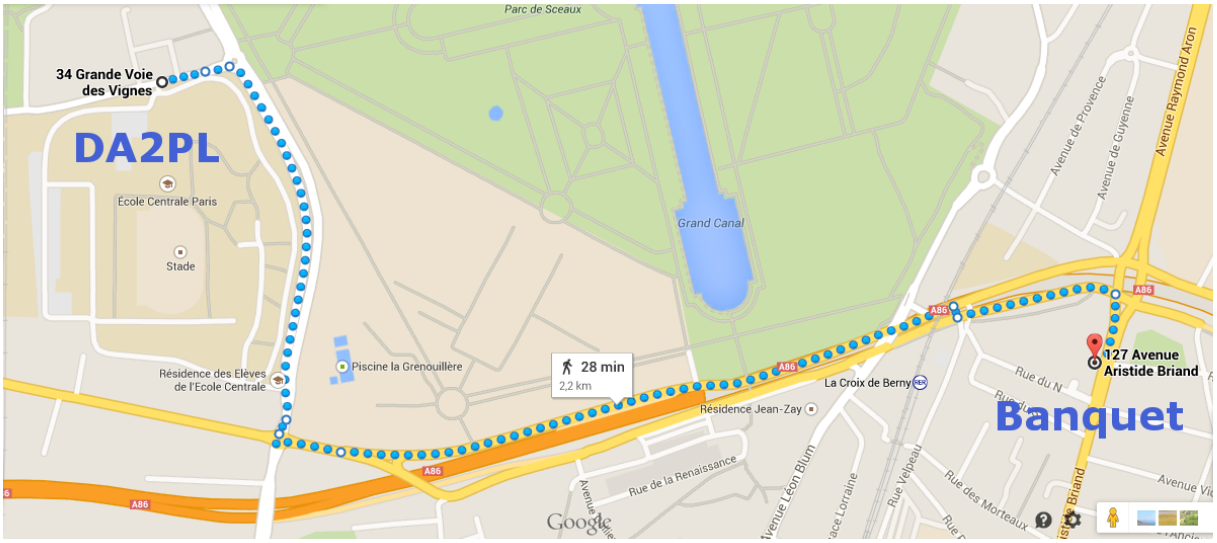
- "*Conjoint axiomatization of the Choquet integral for two-dimensional heterogeneous product sets*",Mikhail Timonin, Queen Mary University of London
  We propose an axiomatization of the Choquet integral model for the general case of a heterogeneous product set $X = X_1 \times X_2$. Previous axiomatizations of the Choquet integral have been given for particular cases $X = Yn$ and $X = \mathbb{R}^n$. The major difference of this paper from the earlier axiomatizations is that the notion of "comonotonicity" cannot be used in the heterogeneous structure as there does not exist a "built-in" order between elements of sets X_1 and X_2. However, such an order is implied by the representation. Our characterization does not assume commensurateness of criteria a priori. We construct the representation and study its uniqueness properties.

- "*Utilitaristic Choquistic Regression*",
  Ali Fallah Tehrani[1], Christophe Labreuche[2], Eyke Hullermeier[1]
  [1]Department of Mathematics and Computer Science, University of Marburg,
  [2] Thales Research & Technology
  Traditionally in machine learning, the attributes are a priori normalized (standardized) and their normalization is not part of the learning process. Taking inspiration from multi-criteria decision aid, we investigate in this paper the interest of learning also the utility function. More specifically we extend two classification methods - namely logistic regression and Choquistic regression - to learn both the normalization and the aggregation of the criteria. Some premilinary results are presented in this paper.

- “*About the french hospitals rankings: a MCDA point of view*”,
  Brice Mayag, LAMSADE, Université Paris Dauphine
  The aim of this paper is to convince the Multi-Criteria Decision Aid (MCDA) and Preference Learning communities to investigate and to contribute in the development of methodologies dedicated to hospital ranking. To do so, we present the french hospital ranking and show how these rankings can be built properly through two existing methods: decision tree and ELECTRE Tri.

19h00 Workshop Banquet

- at Restaurant "*Le Berny*", 127 Avenue Aristide Briand, 92160 Antony, tel.: 01 42 37 72 40



Friday November 21rst, 2014

9h Session 6

- Invited speaker: “*Scaling Optimization Methods for Data-driven Marketing*”,
  Craig Boutillier, University Toronto, Canada ,

  The emergence of large-scale, data-driven analytics has greatly improved the ability to predict the behavior of, and the effect of marketing actions on, individual consumers. Indeed, the potential for fully personalized "marketing conversations" is very real. Unfortunately, advances in predictive analytics have significantly outpaced the ability of current decision support tools and optimization algorithms, precisely the tools needed to transform these insights into marketing plans, policies and strategies. This is especially true in large marketing organizations, where large numbers of campaigns, business objectives, product groups, etc. place competing demands on marketing resources—the most important of which is customer attention. In this talk, I will describe a new approach, called dynamic segmentation, for solving large-scale marketing optimization problems.

  We formulate the problem as a generalized assignment problem (or other mathematical program) and create aggregate segmentations based on both (statistical) predictive models and campaign-specific and organizational objectives. The resulting compression allows problems involving hundreds of campaigns and millions of customers to be solved optimally in tens of milliseconds. I’ll briefly describe how the data-intensive components of the algorithm can be distributed to take advantage of modern cluster-computing frameworks. I will also discuss how the platform supports real-time scenario analysis and re-optimization, allowing decision makers to explore tradeoffs across multiple objectives in real-time.

  Time permitting, I’ll hint at how the technique might be extended to solve sequential, stochastic problems formulated as Markov decision processes, and briefly mention other potential applications of this class of techniques.

10h00 Session 7

- “*Factorization of large tournaments for the median linear order problem*”,
  Alain Guénoche, Institut de Mathématiques de Marseille (I2M - CNRS)
  Computing a median linear order for a given set of linear orders on n elements, is an ordinary task for preference aggregation. This problem is formalized by a tournament (complete directed graph) with n vertices, arcs corresponding to majority preferences. To build a median linear order is to make it transitive, realizing a minimum number of arc-reversal operations. They define the remoteness of any median linear order to this tournament. The computation of a minimum series of arc reversals is usually made using a Branch & Bound algorithm which cannot be applied when n overpasses a few tens. In this text we try to decompose a large tournament (n > 100) into sub-tournaments and to assemble the median orders on each one into a linear order on n elements. We show, making several simulations on random tournaments, weighted or unweighted, that this decomposition strategy is efficient.

- "*Listing the families of Sufficient Coalitions of criteria involved in Sorting procedures*",
  Eda Ersek Uyanik[1], Olivier Sobrie[1,2], Vincent Mousseau[2] and Marc Pirlot[1]
  [1] MATHRO, UMONS, [2] LGI, Ecole Centrale Paris
  Certain sorting procedures derived from ELECTRE TRI such as MR-Sort or the Non Compensatory Sorting (NCS model) model rely on a rule of the type: if an object is better than a profile on a "sufficient coalition" of criteria, this object is assigned to a category above this profile. In some cases the strength a coalition can be numerically represented by the sum of weights attached to the criteria and a coalition is sufficient if its strength passes some threshold. This is the type of rule used in the MR-Sort method. In more general models such as Capacitive-MR-Sort or NCS model, criteria are allowed to interact and a capacity is needed to model the strength of a coalition. In this contribution, we want to investigate the gap of expressivity between the two models. In this view, we explicitly generate a list of all possible families of sufficient coalitions for a number of criteria up to 6. We also categorize them according to the degree of additivity of a capacity that can model their strength. Our goal is twofold: being able to draw a sorting rule at random and having at disposal examples in view of supporting a theoretical investigation of the families of sufficient coalitions.

11h00 Coffee break

11h30 Session 8

- Invited speaker: "*Surrogate loss functions for preference learning*",
  Krzysztof Dembczynski,
  Poznan University of Technology, Poland,
  In preference learning we use a variety of different performance measures to train and test prediction models. The most popular measures are pairwise disagreement (also referred to as rank loss), discounted cumulative gain, average precision, and expected reciprocal rank. Unfortunately, these measures are usually neither convex nor differentiable, so their optimization becomes a hard computational problem. However, instead of optimizing them directly we can reformulate the problem and use surrogate or proxy loss functions which are easier to minimize. A natural question arises whether optimization of a surrogate loss provides a near-optimal solution for a given performance measure. For some of the performance measures the answer is positive, but in the general case the answer is rather negative. During the tutorial we will discuss several results obtained so far.

12h30 Lunch

13h20 Poster session

- "*An Arrow-like theorem over median algebras*",
  Miguel Couceiro[1] and Bruno Teheux[2],
  1 LAMSADE, Université Paris-Dauphine,
  2 Université du Luxembourg
  We present an Arrow-like theorem for aggregation functions over convervative median algebras. In doing so, we give a characterization of conservative median algebras by means of forbidden substructures and by providing their representation as chains.

- "*A Metaheuristic Approach for Preference Learning in Multi-Criteria Ranking based on Reference Points*",
  Jinyan Liu, Wassila Ouerdane, Vincent Mousseau, LGI, Ecole Centrale Paris
  In this paper, we are interested by an aggregation method called multi-criteria ranking method based on Reference Points (RMP). Briefly, instead to have pairwise comparisons between alternatives, the pairs of alternatives are judged according to the reference points. The introduction of such points facilitates the comparison of any two alternatives in which dominance relationship does not necessarily exist. However, we notice that little attention has been brought on how to learn the parameters of this kind of model. Therefore, to tackle this problem, we propose in this work a methodology for preference learning for the RMP method. More precisely, we are interested by learning the parameters of this method when DMs provide us a large set of data or information. Specifically, an algorithm is provided that is a combination of an evolutionary approach and a linear programming approach. Experimental tests and analysis are also presented.

- "*Inferring the parameters of a majority rule sorting model with vetoes on large datasets*",
  Alexandru-Liviu Olteanu, Patrick Meyer, Telecom Bretagne
  The article is centered on the problem of inferring the parameters of a majority rule sorting model when large sets of assignment examples are considered. Beside the proposal of an approach for solving this problem, the main focus of the paper lies in the inclusion of veto thresholds inside the majority rule model, which, as we illustrate, increases the expressiveness of the model. However, due to its complexity, an exact approach for inferring its parameters is not practical especially when large datasets are considered. Therefore, we propose a metaheuristic approach to overcome this difficulty. The approach is validated over a set of constructed benchmarks as well as on several datasets containing real data.

- "*A Dataset Repository for Benchmark in MCDA*",
  Antoine Rolland and Thi-Minh-Thuy Tran, Lab. ERIC, Université Lyon 2
  Several methods have been proposed in the past decades to deal with Multicriteria Decision Aiding (MCDA) problems. However, a comparison between these methods is always arduous as there is no benchmark in this domain. In the same time, people proposing new MCDA methods have no standardized data to deal with to prove the interest of their methods.We propose the creation of a web MCDA DataSet Repository to face this lack of data. We detail the presentation of this repository in this paper.

- "*User Experience Driven Design of MCDA Problems with DecisionCloud*",
  Michel Zam[1,2], Meltem Ozturk[2] and Brice Mayag[2],
  1 KarmicSoft Research,
  2 LAMSADE, Université Paris-Dauphine
  Incremental transformation of stakeholder's decision problems in robust models remains a challenging and complex task that needs better tools. Realistic user experience gives the most valuable input but usually requires several life-cycles. This takes too long, costs too much, and lets precious ideas die. Sketching tools are too superficial, formal modeling tools are too cryptic and development tools are not productive enough.

  We address the evolution vs. consistency challenge and provide an agile solution approach through the whole collaborative modeling process of multicriteria decision problems, including sketching, modeling and interacting with running apps. DecisionCloud is a MCDA extension of the MyDraft platform. Way beyond declaring criteria, alternatives, constraints, evaluations and run classical decision problems, DecisionCloud provides features as domain modeling and instant GUI prototyping. The whole evolutionary process runs in the cloud and is fully traced. Users, designers, and coders, if any, collaborate consistently using only their web browsers and grow their decision models directly in the cloud.

14h00 Session 8

- Invited speaker: "*Preference modeling with Choquet integral*",
  Michel Grabisch, Université Paris 1
  In this talk, we show how capacities and the Choquet integral emerge as natural ingredients when building a multicriteria decision model, especially when the criteria cannot be considered as independent. To face the complexity of the model, we provide efficient submodels based on k-additive capacities, which are naturally connected with the interaction indices, quantifying the interaction existing among criteria in a group of criteria. The case of 2-additive capacities seems to be of particular interest, since it leads to a model which is convex combination of an additive model and max and min over any pair of two criteria. Lastly, we address the issue of the identification of the model through learning data and preferences.

15h00 Coffee break

15h30 Session 9

- "*Characterization of Scoring Rules with Distances: Application to Clustering of Rankings*",
  Paolo Viappiani, LIP6, Université Pierre et Marie Curie
  We consider the problem of clustering rank data, focusing on distance-based methods. Two main steps need to be performed: aggregating rankings of the same cluster into a representative ranking (the cluster's centroid) and assigning each ranking to its closest centroid according to some distance measure. A principled way is to specify a distance measure for rankings and then perform rank aggregation by explicitly minimizing this distance. But if we want to aggregate rankings in a specific way, perhaps using a scoring rule giving more importance to the first positions, which distance measure should we use?
  Motivated by the (known) observation that the aggregated ranking minimizing the sum of the Spearman distance with a set of input rankings can be computed efficiently with the Borda rule, we build a taxonomy of aggregation measures and corresponding distance measures; in particular we consider extensions of Spearman that can give different weights to items and positions

- "*An interactive approach for multiple criteria selection problem*",
  Anil Kaya[1], Özgür Özpeynirci[1], Selin Özpeynirci[2],
  [1] Izmir University of Economics, Department of Logistics Management,
  [2] Izmir University of Economics, Industrial Engineering Department
  In this study, we develop an interactive algorithm for the multiple criteria selection problem that aims to find the most preferred alternative among a set of known alternatives evaluated on multiple criteria. We assume the decision maker (DM) has a quasiconcave value function that represents his/her preferences. The interactive algorithm selects the pairs of alternatives to be asked to the DM based on the estimated likelihood that an alternative is preferred to another one. After the DM selects the preferred alternative, a convex cone is generated based on

this preference information and the alternatives dominated by the cone are eliminated. Then, the algorithm updates the likelihood information for the unselected pairwise questions. We present the algorithm on an illustrative example problem.

- "*FlowSort parameters elicitation: the case of partial sorting*",
  Dimitri Van Assche, Yves De Smet,
  CoDE, Université libre de Bruxelles
  We consider the context of partial sorting. We address the problem of finding the parameters of the FlowSort method using an existing categorization. This contribution constitutes an extension of a method we have developed in the context of complete sorting. It relies on the use of a dedicated Genetic Algorithm based on variations of search parameters. We show how to manage the problem of correct categorization prediction, which is more difficult, since ranges of categories are considered. The method is tested on three different datasets for which a partial sorting has been generated with a particular instantiation of FlowSort.

- "*On confident outrankings with multiple criteria of uncertain significance*",
  Raymond Bisdorff, University of Luxemburg
  We develop Monte Carlo simulation techniques for taking into account uncertain criteria significance weights and ensuring an a priori level of confidence of the Condorcet outranking digraph, depending on the decision maker. Those outranking situations that cannot be ensured at a required level of confidence are assumed to be indeterminate. This approach allows us to associate given confidence degree to the decision aiding artifacts computed from a bipolarly-valued outranking, which accounts for the essential and unavoidable uncertainty of numerical criteria weights.

17h30 Closing session

Group Photo

DA2PL '2014
from Multiple Criteria Decision Aid to Preference Learning
CENTRALE
PARIS

Session 1

- Invited speaker: "*Preference Learning: Machine Learning meets MCDA*"
  Eyke Hüllermeier, Department of Computer Science, Universität Paderborn, 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, which are 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 provide a brief introduction to the field of preference learning and, moreover, to elaborate on its connection to multiple criteria decision aid.

Session 2

- "*On the use of copulas to simulate multicriteria data*", Jairo Cugliari, Antoine Rolland, Thi-Min-Tuy Tran, Lab. ERIC, Université Lyon 2
- "*Data Generation Techniques for Label Ranking*", Massimo Gurrieri, Philippe Fortemps, Xavier Siebert, Marc Pirlot, Nabil Aït-Taleb, MATHRO, Faculté Polytechnique, UMONS

On the use of copulas to simulate multicriteria data

Jairo Cugliari [1] and **Antoine Rolland** [2] and **Thi-Min-Tuy Tran** [3]

Abstract. Several methods have been proposed in the past decades to deal with Multicriteria Decision Aiding (MCDA) problems. However, a comparison between these methods is always arduous as the number of datasets proposed in the literature is very low. One of the limitations of the existing datasets is that generally MCDA method are dealing with very small sets of data; typically, a MCDA problem deals with a number of alternatives that does not exceed 20 or 30 and is often less. Therefore, it should be interesting to propose a way to simulate new data based on some existing dataset, i.e. taking into account the potential links that should exist between the criteria. We introduce in this paper the use of the statistical functions named copula to simulate such data. A practical way to use copula is proposed, and the quality of the obtained data is discussed.

1 Introduction

Multicriteria Decision Aiding (MCDA) studies aim at helping a Decision Maker (DM) to take (good) decisions. Many different models have been proposed since more than 50 years (see [3] or [7] for a survey), among others:

- utility-based approaches, using linear (MAUT [15], AHP [24]) or non-linear (Choquet integral [14]) aggregation functions
- outranking approaches, like ELECTRE [13] or PROMETHEE [8] methods
- mixed methods, like rule-based methods [20, 21] and others.

There is still a great increase of the number of very specific methods, or variants of existing methods, to be proposed. All these methods are always presented as very interesting and perfectly adapted to the situation. The fact is that it is very difficult to test and compare different methods described in the literature, as they often are dedicated to one specific situation. Even if the axiomatic foundations have been generally well studied (see [7] for a first approach), it is often difficult to realize which are the difference *in practice* between the results obtained by two different methods. Therefore, there is a lack of testing sets of data on which one can try the different methods. Several solutions have already been proposed to increase the possibility of benchmark between MCDA methods. We can cite the Decision Deck project which proposes a unified data standard for MCDA data [5], and a unified web services platform through DIVIZ [18]. We can cite also a companion paper [22] which aims at proposing a repository of real or fictitious datasets for MCDA situations.

But sometimes only very few data are available; for example, from an preference learning point of view, the dataset should be so limited that it is too small to be divided into a test subset and a validation subset. Researches should also desire to have more data to test the proposed methods. There is then a need to be able to increase the size of the datasets through simulated data. Good practices in MCDA point out the fact, among others, that criteria should be as independent as possible [23]. But in real life the values taken by an alternative on different criteria are generally not totally independent. For example, if the DM is facing a problem like flat rental, she would like to select several flats to visit. Obviously, data like surface, price, or rooms number seem to be good criteria to decide which flat to visit. But these criteria are often linked: increasing the surface is greater increase also the chance to have more rooms; or the price is an increasing function of the surface, with respect to other criteria. Therefore, MCDA data cannot be independently well simulated. The problem is then to model the interaction between criteria in a plausible way. We propose to use a statistical approach to overcome this difficulty. Copula is a statistical tool which aims at modelling those interactions. Basically, a copula is a function that describe a multivariate distribution as a function of the marginal univariate distributions. We propose in this paper to use copulas to first model the interactions between criteria, and then to simulate new alternatives. We automatically learn the copula parameters from the actual dataset (used as training set) so as to generate new simulated data sets.

As far as we know, there is no work about the simulation of multicriteria data except a tentative using Bayesian network presented in [2].

In this paper we present a practical way to use copulas to simulate MCDA data inspiring from the work in [12]. In section 2 we introduce the copulas functions and quickly present the most well-known copulas families. In section 3 we first stand the hypothesis under which we worked. We then present a process to elicitate the parameters of the copulas following [1]. Finally, we show some numerical experiments we performed on available MCDA dataset in the literature.

2 Copulas

In this section we recall some basic notions about modeling dependency with copulas (see [19] for a more formal presentation of the subject). The basic construction bricks will be pair copula constructions (PCC) which are assembled together in a vine copula.

2.1 A brief introduction to copulas

In a nutshell a copula is a multivariate cumulative distribution function which has all its margins uniformly distributed on the unit interval. If $U_1, \ldots, U_n; n \geq 2$ are random variables with uniform distribution in $[0, 1]$, then a copula $C : [0, 1]^n \mapsto [0, 1]$ satisfies

$$C(u_1, \ldots, u_n) = P(U_1 \leq u_1, \ldots U_n \leq u_n) \qquad (1)$$

[1] Lab. ERIC, Université Lyon 2, email: Jairo.Cugliari@univ-lyon2.fr

[2] Lab. ERIC, Université Lyon 2, email: Antoine.Rolland@univ-lyon2.fr

[3] Lab. ERIC, Université Lyon 2, email: Thi-minh-thuy.Tran@etu.univ-lyon1.fr

A central result on copulas is Sklar's theorem [25] which allows one to represent any $n-$variate cumulative distribution function $F(x_1,\dots,x_n)$ of the random vector $\mathbf{X} = (X_1,\dots,X_n)$ as

$$F(x_1,\dots,x_n) = C(F(x_1),\dots,F(x_n)), \tag{2}$$

where $F(x_1),\dots,F(x_n)$ are the univariate marginal distribution functions of the vector $\mathbf{X}$. Moreover, this representation is unique if the marginals are absolutely continuous. A converse result is Nelsen's corollary [19] which identifies the copula from the joint and marginal distribution

$$C(u_1,\dots,u_n) = F(F^{-1}(x_1),\dots,F^{-1}(x_n)). \tag{3}$$

Intuitively, the probabilistic structure of the vector $\mathbf{X}$ is the result of coupling the marginal behavior of the components of $\mathbf{X}$ by means of the copula C which has intermediate practical implications. For example, from the observation of n independent and identical realizations $\mathbf{X}_1,\dots,\mathbf{X}_n$ of $\mathbf{X}$, one can estimate the joint multivariate distribution function F by estimating the marginals and identifying one copula function among the elements of known copula families (e.g. the elliptical or Archimedean classes among others [19]). If F is absolutely continuous, then we use the chain rule to write the density equivalent to equation (2)

$$f(x_1,\dots,x_n) = c(F_1(x_1),\dots,F_n(x_n))f_1(x_1)\dots f_n(x_n) \tag{4}$$

where the copula density function c is given by

$$c(u_1,\dots,u_n) = \frac{\partial^n C(u_1,\dots,u_n)}{\partial u_1,\dots,\partial u_n} \tag{5}$$

The difficulty of this problem depends on the data dimension n. In the bivariate case, e.g. $n = 2$, only one pair-copula must be estimated and many solutions have been already proposed to do so (see for example [16, Chapter 5]). However, several of these approaches are not feasible in higher dimension spaces.

2.2 Pair-Copula Construction (PCC)

To avoid some problems that arise on high dimension datasets, [4] propose a pair-copula construction in order to decompose the multivariate joint density of X into a cascade of building blocks called pair-copula.

As before f is the joint density of X which is factorized (uniquely up to a relabeling of the elements of X) as

$$f(x_1,\dots,x_n) = f(x_n)f(x_{n-1}|x_n)\dots f(x_1|x_2,\dots,x_n). \tag{6}$$

Then, one can write each of the conditional densities on (6) using (4) recursively which yields on this general expression for a generic element X_i of X given a generic conditioning vector v

$$\begin{aligned} f(x_i|v) =& c_{x_i,v_j|v_{-j}}\left(F(x_i|v_{-j}), F(v_j|v_{-j})\right) \\ & \times f(x_i|v_{-j}). \end{aligned} \tag{7}$$

In last expression we use the notation v_j for the j-th element of v and v_{-j} for all the elements of v but v_j.

For example, let take three random variables X_1,X_2 and X_3. We have the following decomposition:

$$\begin{aligned} f(x_1|x_2x_3) =& c_{12|3}\left(F(x_1|x_3), F(x_2|x_3)\right) \\ & \times f(x_1|x_3). \end{aligned} \tag{8}$$

2.3 Vines copulas

Vines copulas have been proposed to classify alternatives factorization of (6) into a structured graphical model [4]. This construction allows highly flexible decompositions of the (possibly high) dimensional distribution of X because each pair-copula can be chosen independently from the others. The iterative decomposition provided by the PCC is then arranged into a set of linked trees (acyclic connected graph). Two special schemes are usually used: C-vines (canonical vines) and D-vines. In the former one, a dependent variable is identified and chosen to be the root of the tree. In the following tree, the dependence will be computed conditional on this first variable and so on. In the latter scheme, a variable ordering is chosen. Then on the first tree one models the dependence of each of the consecutive pairs of variables. The following tree will model the dependence of the remaining pairs, conditional on the those that were already modeled. See [1] for a more detailed exposition of this construction.

2.4 Simulation

Simulation of copula data (i.e. n-variate data with uniformly distributed marginals) can be done using the probability integral transform. It is convenient to define the h-function

$$h(x|v,\theta) = \frac{\partial^d C_{x,v_j|v_{-j}}(F(x|v_j), F(x|v_{-j}), |\theta)}{\partial F(v_j|v_{-j})}, \tag{9}$$

where θ is a parameter vector associated to the decomposition level. The h-function is the conditional distribution of x given v and we let $h^{-1}(u|v,\theta)$ be its inverse with respect to u, i.e. the inverse of the cumulative conditional distribution. The simulation for the vine is as follows. First sample n uniformly distributed random variables $w_1, w_2, \dots, w_n$. Then use the probability integral transform of the corresponding conditional distribution:

$$\begin{aligned} x_1 &= w_1, \\ x_2 &= F^{-1}(w_2|x_1), \\ x_3 &= F^{-1}(w_3|x_1, x_2), \\ &\dots \\ x_n &= F^{-1}(w_{n-1}|x_1,\dots,x_{-1}). \end{aligned}$$

At each step, the computation of the inverse conditional distribution is made through the (inverse) h-function.

3 Numerical experiments

The aim of the data simulation is to obtain new fictitious data in accordance with a set of real data. The model (copula) parameters are automatically learned from the real dataset, and then the model is used to simulate new data. Ideally, the new fictitious data should be indiscernible from the real ones. We detail in the following sections the hypothesis on the real data that we make, then the simulation process and the way we can prove that we reach our objective of indiscernibility.

3.1 Hypothesis

The input data are a set of p alternatives described on n criteria. Typically, a MCDA problem faces a small number of alternatives (from

5 or 6 to less than 50). The number of criteria is also small ranging between 3 and about 10. It should be noticed that the real data can be considered as *example* data but not as *sampled* data as in the classical statistical sampling theory framework: the data set is not obtained by a random sampling, as the data has been generally previously selected for their interest. Therefore it is difficult to infer the distribution of each criteria from the data, as there exists a observation bias.

Since the margins are unknowns, it is preferable to use normalized ranked data to estimate the copula parameters. This avoids the problem of estimating the marginal distribution. However, we need to estimate these distributions in order to transform the simulated data (whose margins are uniformly distributed) into the original scale of the data. Two different solutions can be considered:

- choose a parametric form of distribution (Gaussian, uniform...) for the criteria and estimate its parameters, or
- use a non-parametric approach for the marginal distribution.

We chose to use the empirical distribution invert function which is a fully non-parametric approach. The inconvenience stands in the fact that we can only infer marginal distribution contained between the observed (real) minimum and maximum for each criterion. Therefore extrema values could be not so well simulated.

In order to avoid problems due to count data we assume that the margins are absolutely continuous. Thus, the representation in (2) is unique.

3.2 Simulation scheme

We use the statistical software `R` to perform the numerical experiments. The simulation process has been implemented in the `CDVine` package [9]. The input data set is a numeric performance matrix. To obtain a simulated dataset we follow these steps:

Step 1. Transform original data into copula data, i.e. purely ordinal distributions for each criterion.

Step 2. Select a C or a D vine structure via the function `CDVineCopSelect` proposed in the package `CDVine`. Parameters of this function are the choice between C-Vine or D-Vine structure to be selected, and the selection criterion (AIC or BIC).

Step 3. Estimate the parameters of all the pair copula jointly through the maximization of the pseudo likelihood. This step is performed via the function `CDVineMLE` proposed in the package.

Step 4. Simulate the desired number of data via the function `CDVineSim` proposed in the package `CDVine`.

Step 5. Transform back copula data into real-like data via the inverse of the empirical cumulative function.

3.3 Evaluation

The testing step consists in the analysis of the differences between the set of real data and the set of simulated one. We want to detect if there is any difference between both sets and quantify the difference if any. An acceptable simulation procedure would yield on simulated data that is indistinguishable from the real data.

Since we are interested on a joint multivariate probability structure, using classical univariate tests (e.g. Kolmogorov-Smirnov test) on the margin of the joint distribution is clearly not sufficient. However, the simulation scheme must warranty that these margins are correctly simulated as well as the joint structure.

One could then rely on clustering methods to split the mixed datasets of real and simulated data into two clusters. Then, one computes a confusion matrix using the classes obtained from the clustering methods and the real labels (real *vs* simulated) and tests for independence through a χ^2 test. The k-means method is one of the most common and popular unsupervised clustering method. However, this method should be useless here, as it will always conclude to the confusion of real and simulated data as long as the marginal distributions will be close. This clustering method is able to capture clusters that are not in the same place in the possible data space, but is less able to capture clusters that have different structures in the same subspace.

Alternatively, one could use a binary classifier to test whether the merged data is easy to discriminate in terms of the added labels `real` *vs.* `simulated`. We use the Random Forest algorithm [10] as a supervised learning method. This algorithm allows to estimate the in-fit sample error rate, that is the proportion of alternatives that are wrongly classified. For this, the algorithm constructs many binary trees classifiers on bootstrapped subsets of the merged dataset. Then, it test the classifier on the remaining alternatives and computes the error rate. The quality indicator we look at is the mean global error ratio computed over all the classifiers constructed by the Random Forest algorithm. Heuristically, the higher the ratio the better it is in our case as it indicates that there is more and more confusion between real and simulated data.

A more formal way of measuring the quality of the simulation is to test the existence of differences between the simulated and real data. For instance, one could use a flexible multivariate ANOVA [26]. We do not explore this method in this paper.

3.4 Results

We tested the elicitation process on 3 data sets obtained from the MCDA data set repository [22]. Let us present the three selected cases.

Case 1. A data set of farms evaluated on the animal welfare, described in [6]. The dataset is composed of 29 farms described on 4 criteria valued between 0 and 100.

Case 2. A data set of 27 scenarios for radioactive waste management, with regard to 4 criteria, described in [11].

Case 3. A data set of 243 virtual peach ideotypes with regard to 3 criteria described in [17].

The three data sets are represented on the left panels of Figures 1, 2 and 3 respectively. These panels contain all the pairwise scatter plots for each data set on its upper triangle. On the lower triangles we represent each estimated pairwise copula density by means of contour plots. These pair copula are the elemental brick on the construction of the vines. It is possible to remark different kinds of probability structures and dependence between the three cases. For instance, the contour plots show spherical shapes in Figure 1 and elliptical shapes in Figure 2 which can be associated to multivariate normal or t distributions. The shapes of the contour plots in Figure 3 are more intricate and therefore represent more complex dependence structures.

In order to obtain a estimation of the quality of the simulation procedure we repeated the simulation scheme (see 3.2) 1000 times, producing then 1000 simulated data sets for each of the three real data sets. The simulated data sets have the same dimensions as the real data sets they are simulated from. Figures 1, 2 and 3 allow to visually inspect one of the replicates of the simulation procedure for each of the MCDA data sets. On the first two cases, it is hard to tell

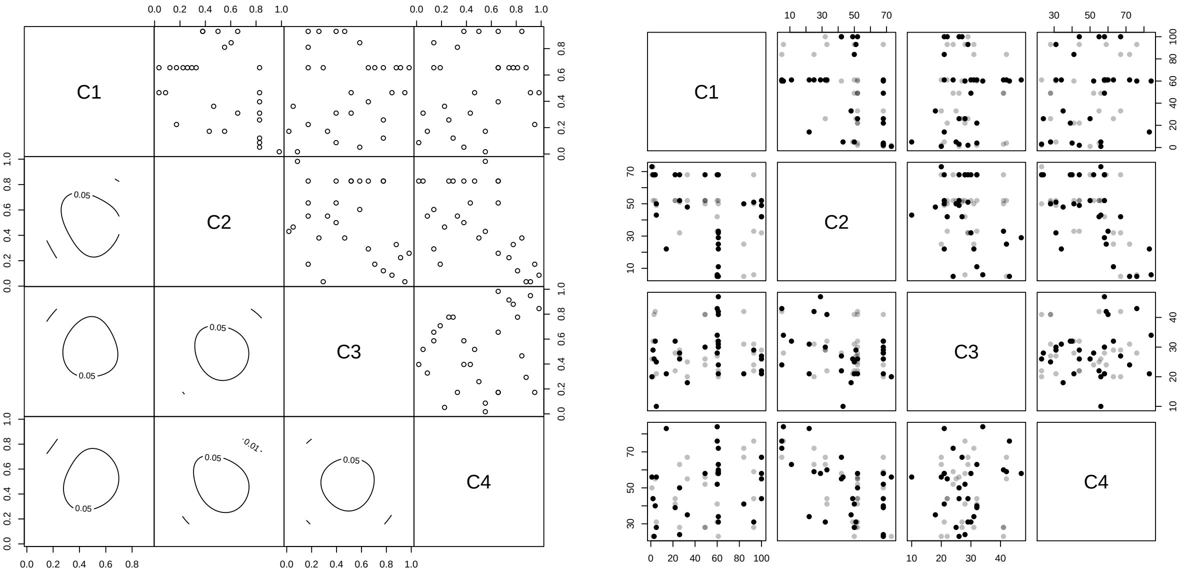


Figure 1: Animal welfare dataset [6]. On the right, the pairs plot of the data points (black dots) and simulated points (gray dots). On the left, the original data points are represented as normalized ranks on the upper triangle and the estimated pair copulas as shown as contour plots on the lower triangle.

that the global pattern of the data is not respected, however there are some simulated data points that lay in zone of low real data density. On the third case, it is more clear that the structure of the simulated dataset does not necessarily follow the structure of the real data. In particular, we see how simulated data points lay far away from the strong structure of real data set.

For each simulation we tested the univariate and multivariate adequacy of the simulated datasets to the real datasets. We used the Kolmogorov-Smirnov test on each margin and the proposed evaluation (see 3.3) using the Random Forest intrinsic error ratio. The Kolmogorov-Smirnov test is marginally rejected at the level of 5% (the maximum rejection was 5 times out of 1000 replication using the first dataset and the first variable) so we do not include the results here. The obtained average error ratios using Random Forest are presented in table 1 which is detailed and analysed in the next section.

In order to ensure that results are not purely due to randomness, we also produced 1000 simulated data sets without any hypothesis of dependence between criteria, i.e. we generated criteria values independently, following only the marginal distributions for each criterion on each dataset. The results of such simulation are also listed in table 1.

	RandomForest Error Ratio	
Data Set	**With copulas**	**Without copulas**
Case 1 [6]	0.534	0.5
Case 2 [11]	0.573	0.012
Case 3 [17]	0.204	0.119

Table 1: Average error ratio of the Random Forest classifier for each of the MCDA datasets for the simulation scheme using copula to model dependence and without any dependence structure.

A higher error ratio shows that it is more difficult to distinguish between learning data and simulated data with the use of copulas, and as a consequence we consider that the simulation is of better quality. If we consider that a higher error ratio implies a simulation of higher quality, then we notice that for all the proposed datasets the quality of the simulated data seems to be better when we use copulas than under the independence hypothesis. However, the differences among the three cases are not negligible. For example while in the first case both error rates are very close (0.534 under the dependence hypotheses against 0.500 under the independence hypothesis), in the second case the error rates are quite different (0.573 under the dependence hypothesis against 0.012 under the independence hypothesis). Let us examine more in detail these results.

First we used vine tree plots to graphically represent the estimated dependence structure. Vine tree plots represent dependent pair copulas as connected components on a graph. The non connected components are the (conditionally) independent couples of variables. When a pairwise dependence exists, the associated edge indicates the strength of the association and a label is placed together with the empirical Kendall's tau as well as the retained copula. Vine tree plots for our experiment are presented on Figure 4. With this representation it is easy to see the low dependence structure of the first case (where only one non independent pair copula is estimated), and also the strong structure observed in case 3 where all the possible pairwise components are linked together. Finally the case 2 is somehow more interesting because the dependence structure is present at some levels of the disaggregation and for some variables.

We now try to interpret the obtained results in terms of the practical problem associated to each one of the cases we studied.

Case 1. [6] presents data that are very weakly dependent (see figure 1). There, the representation of the data and the used copula seems to indicate that the dataset could be correctly represented using a spherical copula. Therefore, a simulation under the independence

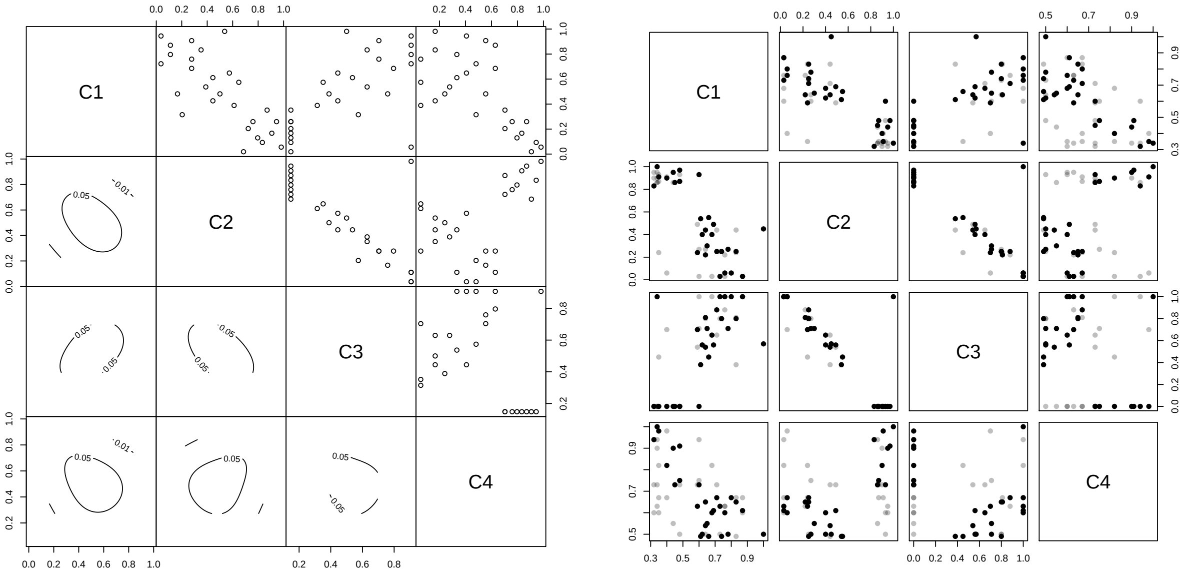


Figure 2: Radioactive waste management dataset. On the right, the pairs plot of the data points (black dots) and simulated points (gray dots). On the left, the original data points are represented as normalized ranks on the upper triangle and the estimated pair copulas as shown as contour plots on the lower triangle.

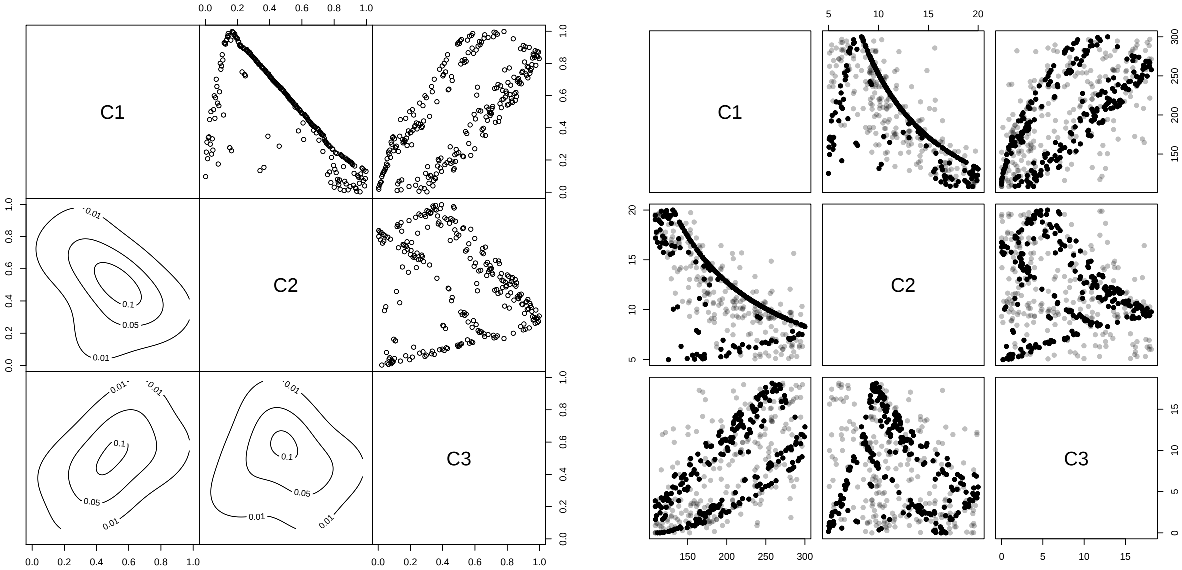


Figure 3: Peach ideotypes dataset. On the right, the pairs plot of the data points (black dots) and simulated points (gray dots). On the left, the original data points are represented as normalized ranks on the upper triangle and the estimated pair copulas as shown as contour plots on the lower triangle.

hypothesis will give good results and will not be very improved by the use of copulas.

Case 2. [11] is a very interesting case as the data are more linked by a non-linear relation (see figure 2 for the representation of the data and the used copula). In this case the use of copulas permits to really improve the quality of the simulation by taking into account these links.

Case 3. [17] is a very special case since all the alternatives in the dataset are situated in a 3D-surface which is a Pareto front (see 3 for the representation of the data and the used copula). Simulating data without the constraint of being in the surface leads for sure to absurd solutions. The use of copulas in this situation can weakly improve the quality of the data, but cannot of course use the special surface structure of the data to better simulate new alternatives. The simulation process should be linked with a cleaning phase where only pareto-optimal solutions should be kept in the dataset.

4 Conclusion

The objective of our work is to propose to the community a practical tool to simulate ”real-like” data from a real dataset. We focused on the way to take into account weak and non-linear links between criteria and proposed a solution based on the use of copulas to model these links. We have shown that the use of copula increases the quality of the simulated data compared to the simple model only based on the use of the marginal distributions. The proposed process is based on a automatic learning of the copula model and parameters. However, we can imagine that the expert can define the used model of copula and/or part of the parameters if needed.

The use of copulas to simulate new MCDA data from a set of real ones seems to be validated by the tests we made. In each case data are of better quality with the use of copulas than if we simulate data under the hypothesis of complete independence between criteria. However, the use of copulas is of higher interest when the criteria are linked by a weak relation: if no relation exists between criteria one can simulate criteria values independently; if a strong (and hidden) relation exists between criteria copulas can fail at representing it.

Moreover, it should be noticed in a multicriteria combinatorial optimization point of view that we only generate alternatives with credible criteria values. We do not check if these alternatives correspond effectively to feasible solutions or not.

Perspectives of this work are the following:

- to provide an efficient similarity-index to test the similarity between real and simulated data;
- to develop an available R service for anyone to simulate MCDA data from a learning dataset;
- to study the effect of criteria number and alternatives number in the learning set on the quality of the simulation;
- to propose a process using copulas to simulate data directly from indications of the DM without any learning dataset.

REFERENCES

[1] K. Aas, C. Czado, A. Frigessi, and H. Bakken. Pair-copula constructions of multiple dependence. *Insurance: Mathematics and Economics*, 44(2):182 – 198, 2009.

[2] N. Ait-Taleb, V. Brison, and M. Pirlot. Generating multicriteria data similar to real data using Bayesian nets. In *11th Decision Deck Workshop*, 2013.

[3] C.A. Bana e Costa, J.M. De Corte, and J.C. Vansnick. On the mathematical foundation of MACBETH. In J. Figueira, S. Greco, and M. Ehrgott, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, pages 409–443. Springer Verlag, Boston, Dordrecht, London, 2005.

[4] T. Bedford and R. M. Cooke. Vines–a new graphical model for dependent random variables. *Ann. Statist.*, 30:1031–1068, 2002.

[5] R. Bisdorff, P. Meyer, and T. Veneziano. XMCDA : a standard XML encoding of MCDA data. In *EURO XXIII : European conference on Operational Research*, pages 53 – 53, 2009.

[6] R. Botreau and A. Rolland. Evaluation multicritère du bien-être animal en ferme : une application des méthodes développées en aide à la décision multicritère. In *8ème congrès de la ROADEF, Clermont-Ferrand*, 2008.

[7] D. Bouyssou, D. Dubois, M. Pirlot, and H. Prade, editors. *Concepts and Methods of Decision-Making*. Wiley-ISTE, 2009.

[8] J.P. Brans and B. Mareschal. PROMETHEE methods. In J. Figueira, S. Greco, and M. Ehrgott, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, pages 163–196. Springer Verlag, 2005.

[9] E.C. Brechmann and U. Schepsmeier. Modeling dependence with c- and d-vine copulas: The r package cdvine. *Journal of Statistical Software*, 52(3):1–27, 2013.

[10] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[11] Th. Briggs, P.L. Kunsch, and B. Mareschal. Nuclear waste management: An application of the multicriteria {PROMETHEE} methods. *European Journal of Operational Research*, 44(1):1 – 10, 1990.

[12] L. Dalla Valle. Official statistics data integration using copulas. *Quality Technology & Quantitative Management*, 11(1):111–131, 2014.

[13] J. Figueira, V. Mousseau, and B. Roy. ELECTRE methods. In J. Figueira, S. Greco, and M. Ehrgott, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, pages 133–53. Springer Verlag, Boston, Dordrecht, London, 2005.

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

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

[16] A. J. McNeil, R. Frey, and P. Embrechts. *Quantitavie Risk Management*. Princeton Series in FInance, 2005.

[17] M.-M. Memmah, A. Rolland, and B. Quilot-Turion. Multicriteria sorting methods to select virtual peach ideotypes. *International Journal of Multicriteria Decision Making*, to appear, 2014.

[18] P. Meyer and S. Bigaret. Diviz: A software for modeling, processing and sharing algorithmic workflows in MCDA. *Intelligent decision technologies*, 6:283 – 296, 2012.

[19] R.B. Nelsen. *An Introduction to Copulas*. Springer, second edition, 2006.

[20] Z. Pawlak. Rough sets. *International Journal of Information and Computer Sciences*, 11:341–356, 1982.

[21] Z. Pawlak. *Rough Sets. Theoretical Aspects of Reasoning about Data.* Kluwer, Dordrecht, 1991.

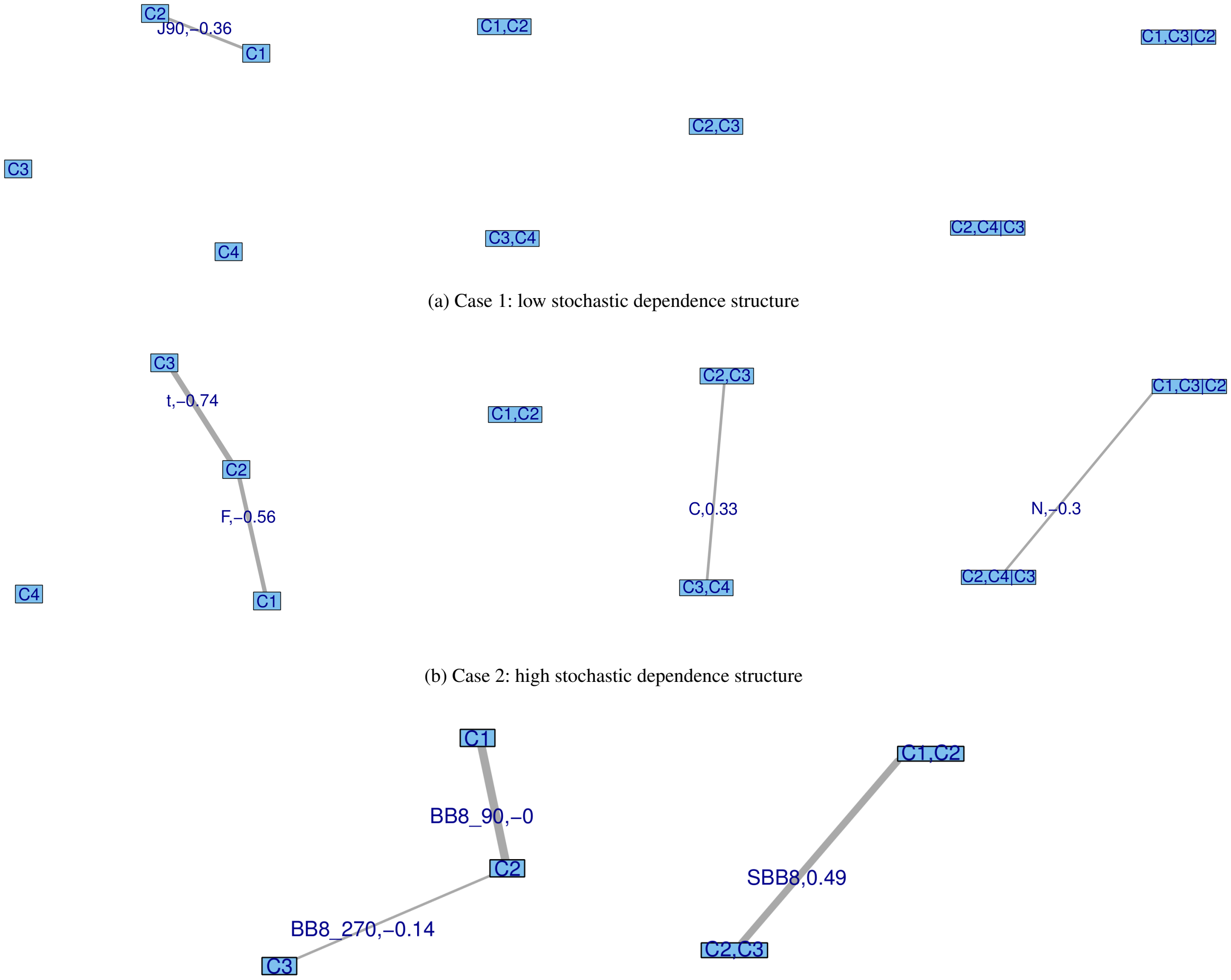
[22] A. Rolland and T.M.T. Tran. A data set repository for benchmark in mcda. In *DA2PL 2014 Workshop From Multiple Criteria Decision Aid to Preference Learning*, page ??, 2014.

[23] B. Roy. *Multicriteria Methodology for Decision Aiding*. Kluwer Academic, Dordrecht, 1996.

[24] T.L. Saary. *The analytic hierarchy process*. McGraw Hill International, New York, 1980.

[25] A. Sklar. Fonctions de répartition à n dimensions et leur marges. *Publications de l'Instiutut de Statistique de l'Université de Paris*, 8:229–231, 1959.

[26] M.A. Zapala and N.J. Schork. Multivariate regression analysis of distance matrices for testing associations between gene expression patterns and related variables. *Proceedings of the National Academy of Sciences*, 103:19430–19435, 2006.



(a) Case 1: low stochastic dependence structure

(b) Case 2: high stochastic dependence structure

(c) Case 3: high deterministic dependence structure

Figure 4: Tree vine plot for the three cases studied. Nodes represent the pair copula components. Edges are present where a significant dependence is estimated between pair copula.

Data Generation Techniques for Label Ranking

Massimo Gurrieri [1], Xavier Siebert, Nabil Aït-Taleb, Marc Pirlot, Philippe Fortemps

UMONS, Université de Mons, Faculté Polytechnique, rue de Houdain 9, 7000 Mons, Belgium

Abstract

In view of the lack of benchmark data for label ranking, experimentations are typically performed on data sets derived either from classification or regression data. However, though such data sets are accepted by several researchers in this field, they do not provide in general trustworthy data and fail to deal with more general settings. Moreover they are not guaranteed to verify more general properties suited for particular settings. This paper proposes datasets generation techniques to provide synthetic data suitable for label ranking and its extensions.

Keywords: Preference Learning, Label Ranking, Machine Learning, Data generation, Bayesian Networks, Multi-criteria Decision Aid.

1 Introduction

The topic discussed in this paper concerns label ranking [1–5], a prediction task in preference learning, where the goal is to learn a map from instances to rankings over a finite set of classes (or labels). The main goal in label ranking is to predict weak or partial orders (more generally total orders) of labels for a new query (or instance). For example, a group of custumers is willing to rank five products, such that position one is associated to the best product (the top ranked one), position two to the second best and so on. The learning of such a model (i.e. the label ranker) is based on custumer's features (e.g. his/her net salary, his/her age, etc.) and will be capable of predicting a ranking on these five products for a new custumer based on his/her features. Several methods to label ranking have been recently presented [1–7] and in view of the lack of benchmark data for label ranking, experimentations were mainly performed on data sets derived either from classification or regression data sets, i.e. from machine learning repositories (UCI, MLData, StatLib, Statlog...) [5]. However, though such data sets are well accepted by several researchers in this field, they do not provide, in general, trustworthy data for label ranking data and fail to deal with more general settings, namely label ranking where the learning set contains either *incomplete rankings* or *partial orders* instead of linear orders. Moreover they are not guaranteed to verify more general properties (e.g. prior knowledge of monotonicities between attributes and labels or correlations between labels) suited for particular settings. The generation of artificial datasets is however not trivial since instances have to be associated with full/incomplete rankings over a finite set of labels and attributes (i.e. the feature vector) have to be linked with such orders. In this paper we discuss some methods to generate artificial data sets for label ranking.

This paper is organized as follows. In section 2, we introduce label ranking more formally. In section 3, we describe existing datasets for labels ranking. In section 4 and 5, we introduce approaches to generate data sets suitable for label ranking. Finally, in section 6 and 7, we present some experimental results and conclusions, respectively.

2 Label Ranking

In label ranking [6, 7] the main goal is to predict for any instance x, from an instance space X, a preference relation $\succ_x$: $X \rightarrow L$, where $L=\{l_1; l_2; ...; l_k\}$ is a finite set of labels or alternatives, such that $l_i \succ_x l_j$ means that instance x prefers label l_i to label l_j or, equivalently, l_i is ranked higher than l_j. More specifically, we are interested to the case where $\succ_x$ is a total strict order over L, or equivalently, a ranking of the entire set L. This ranking can therefore be identified with a permutation $\pi_x \in \Omega$ (the permutation space of the index set of L), such that $\pi_x(i) < \pi_x(j)$ means that label l_i is preferred to label l_j ($\pi_x(i)$ represents the position of label l_i in the ranking).

3 Data sets from Machine Learning

Existing label ranking data sets are available at KEBI Data Repository [2]. These data sets are essentially multiclass and regression data sets from the UCI repository and the Statlog collection that were turned into label ranking data in two different ways. As for classification data (denoted type A), the procedure proposed in [1] consists in training a naive Bayes classifier on the complete data set. For each training instance, all the class/labels present in the data set are then ordered with respect to the predicted class probabilities (ties are arbitrarly broken). As for regression data (denoted type B) [5], a certain number of (numerical) attributes are removed from the set of predictors and are accordingly considered as labels. Finally, to obtain a ranking for each instance, the (removed) attributes are standardized and sorted by decreasing order of their values. Since the overall original attributes are correlated, the remaining predictive attributes will be somehow correlated with the final rankings. As stated in [5], experimentally, the second type of data generation produces more difficult learning problems (in view of the correlation between remaining and removed attributes). However, though the above-mentioned datasets are used for experimentations related to label ranking, they lack of trustworthiness and cannot be used to model more general settings where, for example, the training set contains incomplete rankings or partial orders instead of full rankings. This is mainly due to the fact that even when two labels are incomparable or indistinguishable, they are arbitrarily discriminated and ordered to form a total order. In the case of type-A data, very often, when running the procedure suggested in [1, 5], it is possible to observe that there is a dominating class (the one with the highest class probability) while the other classes are more or less equally distributed. As an example, Table 1 partially shows class probability distributions obtained with a naive Bayes classifier for the multiclass data set Glass (214 instances, 9 attributes and 7 classes) run on Weka [8]. It is evident that (almost) the overall probability is always given to one class, while the other classes have probability values close to zero. Thus, the way rankings are obtained from such probabilities values is somehow unreliable. As a main consequence, different strategies to break ties may lead to different rankings on the set of labels, so that the performances of a label ranker could be directly influenced by such a choice. As for regression data sets, the way labels are chosen among the predictors is once again arbitrary and unclear. Therefore, it is clear that converting classification/regression data into label ranking data cannot always ensure the trustworthiness of data. To sum up, it would be very useful to provide reliable methods for generating synthetic data that would allow one not only to have trustworthy data, but also to simulate/represent more general settings and to take into account more general properties (e.g. monotonicities between attributes and labels [9] or correlations between labels [10, 11]) that could be useful for future research in this field. In view of the above, in this paper, we present methods for the generation of artificial data sets (denoted type C) based

L_1	L_2	L_3	L_4	L_5	L_6	L_7
0	0	0	0	0	0	**1**
0.853	0.048	0.1	0	0	0	0
0	**0.884**	0	0	0.102	0	0.014
0.782	0.213	0.005	0	0	0	0
0.957	0.013	0.029	0	0	0	0
0	0.072	0	0	**0.928**	0	0

Table 1: Partial class **probability distributions** for the (multiclass) data set **Glass** obtained with a naive Bayes classifier. Each row represents the predicted probability distribution of classes L_1-L_7. Such distributions tend to privilegiate one class, while the others are more or less equally distributed with probabilities close to zero.

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

DATA	Type	Instances	Labels	Attributes
DATAGEN1	C	1000	5	10
DATAGEN2	C	1000	5	10
DATAGEN3	C	1000	5	5
DATAGEN4	C	1000	4	5
DATAGEN5	C	1000	4	5

Table 2: Summary of generated synthetic data

on Multi-criteria decision making approaches [12] and on Bayesian Networks [13]. A summary of five exemplary artificial data sets, and their properties, that we generated with the methods proposed in this paper, is given in Table 2.

4 Artificial Data sets: Utility function approach

In this section we present methods for generating synthetic data for label ranking that are based on the notion of utility functions [6,12,14], where it is possible to provide rankings (total orders) among the set of labels based on values of their utilities. Note that, if a threshold for utilities is fixed, then the preference of a label over another one can also be associated with a notion of intensity, meaning that utilities within that threshold of each other are declared incomparable (rejecting the assumption that indifference is transitive, i.e. arising semiorders [12] instead of total orders). Let $L = \{l_1; l_2; ...; l_K\}$ be the set of labels for which a ranking has to be provided based on the n-dimensional feature vector $x = (a_1, a_2, ..., a_n)$ describing an instance $x \in X$, with $|X| = N$. Thus, the final data set will be comprised of instances such as:

$$(x, \pi_x) = (x, l_{\pi_x^{-1}(1)} \succ_x l_{\pi_x^{-1}(2)} \succ_x \dots \succ_x l_{\pi_x^{-1}(K)}) \tag{4.1}$$

In order to generate such a ranking, labels will be ordered w.r.t. decreasing values of (real-valued) utility functions, one for each label:

$$f_k : X \to \mathbb{R} \quad \forall k \in 1, 2, ..., K \tag{4.2}$$

so that $l_h \succ_x l_k \Leftrightarrow \pi_x(i) < \pi_x(j) \Leftrightarrow f_h(x) > f_k(x)$.

In the following three sections, we discuss three different approaches for label ranking data sets generation based on the notion of utilities. Though the presented methods concern the generation of instances associated with strict linear orders, they can easily be adapted to deal with more general settings (partial orders or semiorders instead of linear orders, as discussed above).

4.1 Data sets generation method: DATAGEN1

In our first proposed method, we assume that each label l_k, $k = 1, 2, ..., K$, is characterized by a M-dimensional vector (or equivalently a set of M criteria). Thus, the utility given by the generic instance i, $i = 1, 2, .., N$, to label l_k will be a weighted sum on the M criteria:

$$f_k(i) = \sum_{s=1}^{M} \omega_s^i c_{s,k} \quad (k = 1, 2, ..., K) \tag{4.3}$$

where the weight ω_s^i is linear combination of the evaluations of the instance i on the n attributes and $c_{s,k}$ represents the evaluation of the kth label on the sth criteria:

$$\omega_s^i = \sum_{j=1}^{n} a_{i,j} b_{j,s} \tag{4.4}$$

as explained in the following. Let $A = [(a_{i,j})]$ be the feature matrix wherein the ith line is an n-dimensional vector describing the instance i among a set of N instances, w.r.t. to n attributes (i.e. its feature vector or equivalently, the evaluation of the instance i on each of the n attributes). Thus $i = 1, 2, .., N$ and $j = 1, 2, ...n$. The matrix A, containing the feature vector of each instance, can be randomly generated.

Let us now define the matrix $B = [(b_{j,s})]$, $j = 1, 2..., n$, $s = 1, 2, ..., M$, where the sth column $b_{j,s}$, $j = 1, 2, ..., n$, contains a n-dimensional vector of weights related to the n attributes $a_1, a_2, ..., a_n$ w.r.t. to the current criterion s. In other words, columns $1, 2, ..., s$ can be considered as s different ways of giving importance to the n attributes. Note that the number M of such models can be chosen arbitrarly

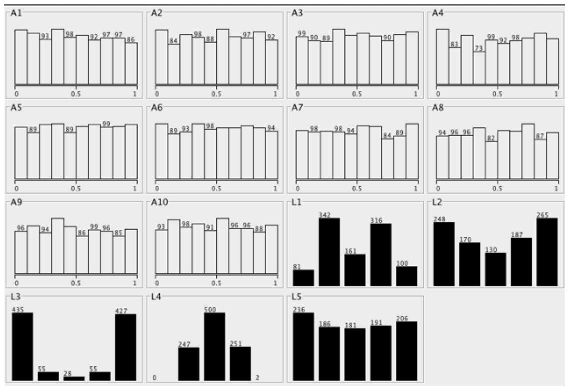


Figure 1: DATAGEN1: Histograms of attributes A1-A10 and of labels L1-L5. $L3$ occupies most of times the first or the last position while $L4$ never occupies the first position and only two times the last position in the generated rankings.

depending on how each label is described. The matrix B is generated with the constraint that the elements of each column sum up to one (they are weights given to attributes):

$$\sum_{j=1}^{n} b_{j,s} = 1 \tag{4.5}$$

In order to draw normalized vectors of weights in uniform way, we used the algorithm discussed in [15] that prevents non-uniformity of sampling.

Let us finally define the matrix $C = [(c_{s,k})]$, $s = 1, 2, ..., M$, $k = 1, 2, ..., K$, where the generic kth column $c_{s,k}$, $s = 1, 2, ..., M$, contains the evaluations of the kth label on the M criteria. The columns of the matrix C are generated similarly as for matrix $A = [(a_{i,j})]$. The output of this method will be a set of instances $(x^i, \pi_{x^i}) = (a_{i,1}, ..., a_{i,n}, \pi_{x^i})$, where the ranking π_{x^i} on labels is obtained by ordering labels w.r.t. values of their utilities (obtained by inserting 4.4 in 4.3):

$$f_k(i) = \sum_{j=1}^{n} \sum_{s=1}^{M} a_{i,j} b_{j,s} c_{s,k} \quad (k = 1, 2, ..., K) \tag{4.6}$$

thus $f_k(i)$ is a weighted sum of the M criteria related to the label k, where weights are linear combinations of the n attributes specific for the ith instance. Figure 1 shows the attributes and labels distributions of the generated data set. Though attributes $A1$ to $A10$ are uniformly distributed (at least approximatively), labels $L3$ and $L4$ present some bias. $L3$ occupies most of times the first or the last position while $L4$ never occupies the first position and only two times the last position in the generated rankings.

4.2 Data sets generation method: DATAGEN2

The method described in this section can be considered as a specific instance of the method DATAGEN1 where the utility given by the generic instance i, $i = 1, 2, .., N$, to label l_k is simply a weighted sum of the n attributes.

Let define the matrix $A = [(a_{i,j})]$ as in the previous section, while $B = [(b_{j,k})]$, $j = 1, 2..., n$, $k = 1, 2, ..., K$ is now a matrix whose the generic kth column $b_{j,k}$, $j = 1, 2, ..., n$, contains weights given to the n attributes used to describe the kth label. In other words, such a column represents the importance given by a generic instance i to the n attributes that describe the kth label. Note that the elements of B are generated as in DATAGEN1. The output of this method will be a set of instances $(x^i, \pi_{x^i}) = (a_{i,1}, ..., a_{i,n}, \pi_{x^i})$, similarly to method DATAGEN1, where the ranking π_{x^i} on labels is obtained by ordering labels w.r.t. their utilities:

$$f_k(i) = \sum_{j=1}^{n} a_{i,j} b_{j,k} \quad (k = 1, 2, ..., K) \tag{4.7}$$

Figure 2 shows attributes and labels distributions of the generated data set. Attributes $A1$ to $A10$ as well as labels $L1$ to $L5$ are uniformly distributed.

4.3 Data sets generation method: DATAGEN3

A slightly different method is described hereinafter. Let $\Omega = [(\omega_{i,j})]$, $i = 1, 2..., N$, $j = 1, 2, ..., n$, be a matrix where each line represents a n-dimensional set

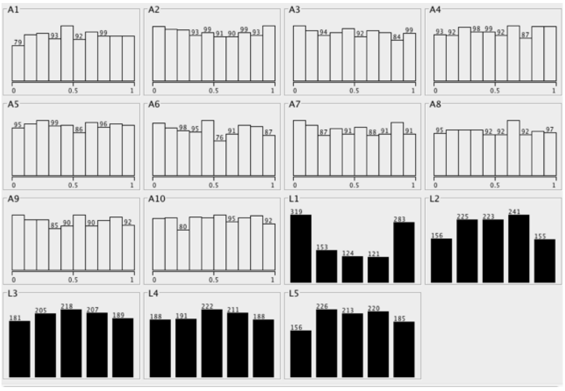


Figure 2: DATAGEN2: Histograms of attributes A1-A10 and of labels L1-L5. All labels are uniformly distributed along the positions of the generated rankings

of weights. Thus, a generic line i identifies a particular set $S_i = (\omega_{i,1}, \omega_{i,2}, ..., \omega_{i,n})$ of weights associated to the n attributes. The main idea is that if we want to rank K objects (i.e. labels) through a weighted sum of their n attributes, different sets of weights should provide different rankings on these objects. Thus, N sets of weights will provide N different rankings on the given objects. Thus the output of this method will be a set of instances $(x^i, \pi_{x^i}) = (S_i, \pi_{S_i}) = (\omega_{i,1}, ..., \omega_{i,n}, \pi_{x^i}), i = 1, 2..., N$ where the ranking π_{S_i} is obtained by ordering labels w.r.t. their utilities:

$$f_k(i) = \sum_{j=1}^{n} \omega_{i,j} \; g_{j,k} \quad (k = 1, 2, ..., K) \tag{4.8}$$

where the rows of the matrix $\Omega = [(\omega_{i,j})]$ are generated as for DATAGEN1 and DATAGEN2 (i.e. the elements of each row are normalized weights) while the generic kth column of the matrix $G = [(g_{j,k})]$ contains the evaluations of the kth label on the n attributes and can be randomly generated. Nevertheless, in order to guarantee monotonicity between attributes and labels, the matrix G could also be obtained, for example, by using a real data set where such a monotonicity is provided. Figure 3 shows the attributes and labels distributions of the generated data set. In this example, we used the car's choice data set [14] to obtain the matrix G. Attributes $A1$ to $A5$ are not uniformly distributed and all labels reveal some bias. $L1$ always occupies the forth and the fifth position, $L2$ and $L3$ always occupy the third, fourth and fifth positions and $L4$ and $L5$ occupies the first and second positions in all generated rankings.

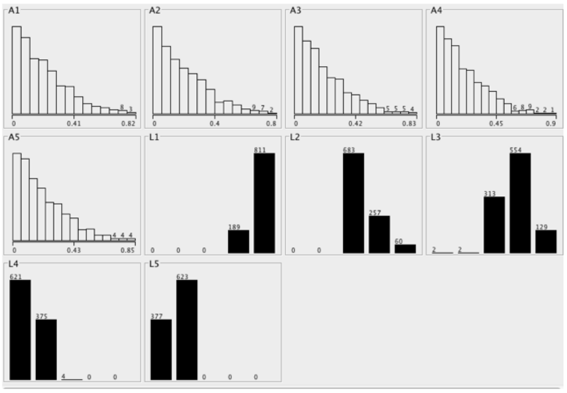


Figure 3: DATAGEN3: Histograms of attributes A1-A5 and of labels L1-L5. Attributes and labels are not uniformly distributed. $L1$ always occupies the forth and the fifth position, $L2$ and $L3$ always occupy the third, fourth and fifth positions and $L4$ and $L5$ occupies the first and second positions in all generated rankings.

5 Artificial Data sets: Bayesian Networks

In previous approaches the feature matrix A, C and G containing, respectively, the evaluations of each instance on attributes, the evaluations of each label on a set of criteria and the evaluations of each label on attributes, are drawn independently, which is to some extent not realistic. In this section, we present a method that allows to generate correlated evaluations. The present method is based on Bayesian Networks [13] and allows to generate data wherein conditional dependencies between attributes and labels can be explicitly given as input.

Assume that we want to generate N-dimensional data vectors whose correlations are as close as possible to a correlation matrix R given as input. In the following, we briefly describe the principles of the method designed for this purpose. However, the interested reader is referred to the working paper [16] for further information.

Let us fix an order on the N dimensions and consider a Bayesian Network whose the graph is a directed acyclic graph (DAG) of this order. The N variables are genereted iteratively as follows. The first random variable X_1 is simply drawn from a standard normal distribution, i.e. $X_1 \sim \mathcal{N}(0, 1)$. The second random variable X_2 is generated in such a way as to preserve the value of the required correlation (given in the matrix R) between the first two variables X_1, X_2, i.e. X_2 is obtained as a noisy linear regression of the variable X_1. The third variable X_3 is similarly obtained so as to preserve the given correlation with X_1 and X_2, and so on. As a result, each variable X_i, $i = 1, 2, ...N$, is obtained as a noisy linear regression on the previuos variable $X_1, X_2, ..., X_{i-1}$. At a given step, in case it is not possible to exactly preserve the specified correlations, a matrix algorithm [17] is used to minimize the (Frobenius) distance between the current correlation matrix and matrix R given as input. At the end of this process, the associated Bayesian Network can generate random vectors distributed according to a multivariate Gaussian distribution whose the correlation matrix is, in a certain sense, the correlation matrix closest to the given matrix R. Though each variable X_i, $i = 1, 2, ...N$, is $\mathcal{N}(0, 1)$, it is possible to adjust the marginal means and variances by using an appropiate affine transformation. Note that such a process can be adapted to the case in which the matrix R is only partially specified. Alternatively, one can specify the main correlations by drawing an acyclic graph. The desired correlations are specified for the arcs of the graph only. In such a case, the order relation on the variables is chosen in such a way as to contain all arcs of the graph (which is always possible since the graph is acyclic). In the current label ranking setting, a specific instance of the desired synthetic dataset can be represented as a directed acyclic graph (DAG) whose nodes are either attributes (in arbitrary number) or labels, while edges represent conditional dependencies between any two nodes. In the following sections, we discuss two exemplary datasets generated by means of the present method.

5.1 DATAGEN4: the case of uncorrelated labels

In the present case, we do not impose correlations between labels evaluations (i.e. the set of labels is an independent set of nodes in the DAG). Thus, all arcs in the DAG are either linking attribute nodes between them or attribute nodes with label nodes. Moreover, in order to guarantee uncorrelated label evaluations, all arcs linking attribute nodes with label nodes should have their origin at label nodes, as shown in Figure 4. As a consequence, label evaluations will be drawn independently. The correlations obtained for DATAGEN4 are represented in Table 3 and are in accordance with the required correlations given as input. The generated data can be interpreted as follows: the sub-vector corresponding to attribute values represent the feature vector of a specific instance while the sub-vector corresponding to the label values represent utilities that are used to form a ranking on labels. Note that label values are uncorrelated and that attribute values and label values follow Gaussian distributions with mean 100 and standard deviation 20. In particular, the attributes values are noisy linear regressions of the labels values.

5.2 DATAGEN5: the case of correlated labels

In this exemplary dataset, DATAGEN5, we imposed some correlations between labels nodes, as depicted in Figure 5. The correlations obtained for DATAGEN5 are represented in Table 4 and are in accordance with the required correlations given as input. Attribute values and label values also follow Gaussian distributions with mean 100 and standard deviation 20. Note that the possibility to establish correlations between labels, as proposed in this paper, would be particularly useful in the context of preference learning and more particularly in multi-label classification where the interdependency (or correlation) between labels is a crucial issue to take into account during the learning phase [10, 11].

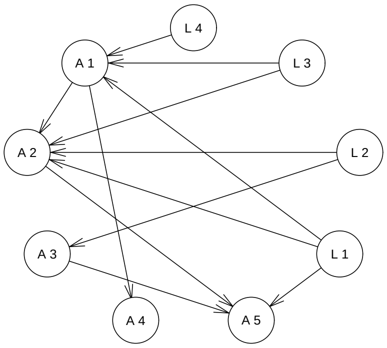


Figure 4: Conditional Dependencies Structure of DATAGEN4.

6 Experimental setup

This section is devoted to experimentations on the artificial datasets generated with the presented methods. The evaluation measures used in this study are the *Kendall's tau* and the *Spearman's rank correlation coefficient* [18]. A cross validation study (10-fold) was performed. The following methods were used in our experimentation: ranking by pairwise comparison (RPC) [1], nominal coding decomposition (ND) [2–4] and random classifier chains for label ranking (CD) [4]. The experimental results, in terms of Kendall's Tau and Spearman's rank corrrelation, are shown in Table 5. In this experiment, we evaluated all algorithms using WEKA [8] in batch mode from a Python program and Radial Basis Function (RBF) as base-classifier with default parameters.

DATAGEN4, DATAGEN5 are the most difficult datasets to learn since both measures are lower w.r.t. any method. Conversely, DATAGEN3, DATAGEN1 and DATEGEN2 are less difficult to learn. At least intuitively, this could be explained by the distributions of attributes and labels, as shown in Figures

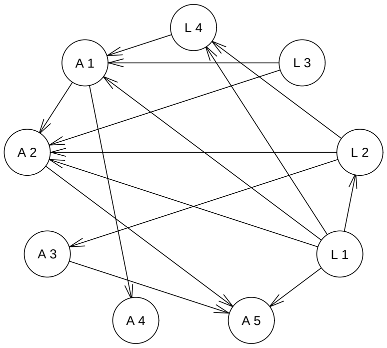


Figure 5: Conditional Dependencies Structure of DATAGEN5.

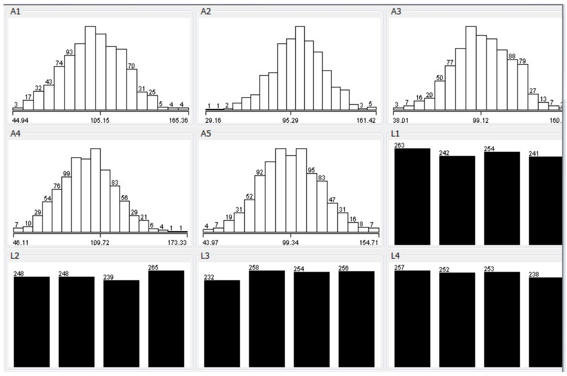


Figure 6: DATAGEN4: Bayesian Network (no correlation between labels). Histograms of attributes A1-A5 and of labels L1-L4. Attributes $A1$ to $A4$ follow (approximately) a Gaussian distribution around the mean while all labels are uniformly distributed along the positions of the generated rankings.

	A_1	A_2	A_3	A_4	A_5	L_1	L_2	L_3	L_4
A_1	1.000	0.631	0.044	0.111	0.052	0.484	-0.003	-0.423	0.405
A_2	0.631	1.000	0.084	0.015	-0.567	0.115	-0.201	-0.525	0.080
A_3	0.044	0.084	1.000	0.062	-0.172	-0.014	-0.388	-0.003	0.052
A_4	0.111	0.015	0.062	1.000	0.001	0.133	-0.035	0.129	0.058
A_5	0.052	-0.567	-0.172	0.001	1.000	-0.058	0.098	0.087	0.133
L_1	0.484	0.115	-0.014	0.133	-0.058	1.000	0.008	0.026	0.015
L_2	-0.003	-0.201	-0.388	-0.035	0.098	0.008	1.000	-0.021	-0.003
L_3	-0.423	-0.525	-0.003	0.129	0.087	0.026	-0.021	1.000	-0.003
L_4	0.405	0.080	0.052	0.058	0.133	0.015	-0.003	-0.003	1.000

Table 3: Correlation matrix obtained for DATAGEN4. Correlation values are in accordance with the correlations given as input.

	A_1	A_1	A_3	A_4	A_5	L_1	L_2	L_3	L_4
A_1	1.000	-0.318	0.002	0.430	0.063	-0.108	-0.030	0.713	0.442
A_2	-0.318	1.000	0.029	0.068	-0.252	0.499	0.349	-0.383	0.152
A_3	0.002	0.029	1.000	-0.002	-0.624	0.078	-0.027	0.020	-0.013
A_4	0.430	0.068	-0.002	1.000	0.010	0.127	0.104	0.115	0.108
A_5	0.063	-0.252	-0.624	0.010	1.000	-0.180	-0.030	0.064	-0.043
L_1	-0.108	0.499	0.078	0.127	-0.180	1.000	0.404	0.013	0.286
L_2	-0.030	0.349	-0.027	0.104	-0.030	0.404	1.000	-0.013	0.516
L_3	0.713	-0.383	0.020	0.115	0.064	0.013	-0.013	1.000	0.048
L_4	0.442	0.152	-0.013	0.108	-0.043	0.286	0.516	0.048	1.000

Table 4: Correlation matrix obtained for DATAGEN5. Correlation values are in accordance with the correlations given as input.

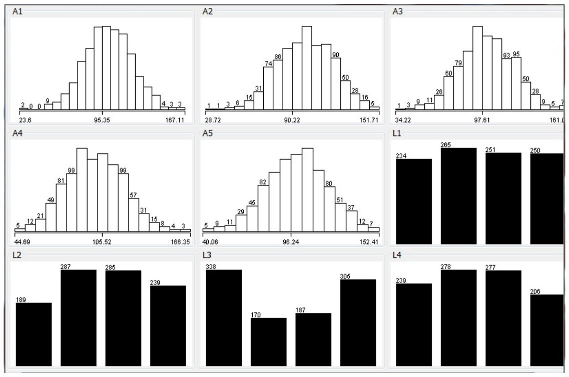


Figure 7: DATAGEN5: Bayesian Network (correlation between labels). Histograms of attributes A1-A5 and of labels L1-L4. Attributes $A1$ to $A4$ follow (approximately) a Gaussian distribution around the mean. All labels are (approximately) uniformly distributed along the positions of the generated rankings except L3 which seems to appear mostly either in the first or the last position of the generated rankings.

1-7: datasets having labels uniformly distributed over the ranking positions (DATAGEN4, DATAGEN5) are more difficult to learn. Conversely, datasets (DATAGEN3, DATAGEN1, DATAGEN2) for which labels are distributed less uniformly are less difficult to learn (i.e. they present a bias due to the fact that some labels occupy very often the same position in the rankings).

7 Conclusions

In this paper, we introduced some techniques for generating datasets suitable for label ranking. We mainly investigated two generation methods: a first method which is based on the concept of utility functions (DATAGEN1, DATAGEN2, DATAGEN3) and a second one which is based on Bayesian Network (DATAGEN4, DATAGEN5). In particular, the latter allows to generate data where some statistical parameters (mean, variance and correlation) can be given as input. In order to study such datasets, we used some label ranking methods and evaluate their performances w.r.t. to Kendall's Tau and Spearman's rank correlation. As a main result, datasets having labels uniformly distributed (DATAGEN4, DATAGEN5 and DATAGEN2) over the ranking positions are more difficult to learn. Conversely, datasets for which some labels present a bias (i.e. they occupy very often the same position in a ranking) are less difficult to learn. In particular, as expected, DATAGEN4 is the most difficult dataset to learn, since labels are independent from attributes and uncorrelated between them. Beside the experimental results provided for the exemplary datasets generated with the proposed methods, this paper discusses and attempts to solve an important issue in label ranking, namely how to generate synthetic data suitable for this setting. The methods proposed in this paper would be useful not only to provide trustworthy label ranking data, but also to simulate/represent more general settings (incomplete rankings, partial orders) and to take into account more general properties required for future research in this field.

Kendall tau

	SD	ND	CC
DATAGEN1	.794+-.020	.427+-.073	.724+-.035
DATAGEN2	.750+-.021	.486+-.028	.704+-.024
DATAGEN3	.973+-.008	.895+-.018	.960+-.011
DATAGEN4	.252+-.057	.168+-.005	.220+-.007
DATAGEN5	.322+-.036	.242+-.065	.297+-.004

Spearman's rank correlation

	SD	ND	CC
DATAGEN1	.861+-.018	.512+-.083	.776+-.036
DATAGEN2	.835+-.019	.588+-.027	.798+-.022
DATAGEN3	.986+-.004	.946+-.009	.979+-.006
DATAGEN4	.313 +- .063	.200 +- .073	.267 +- .008
DATAGEN5	.379 +- .037	.285 +- .075	.348+-.004

Table 5: Kendall's Tau and Spearman's rank correlation on Artificial Data Sets - RBF as base classifier

References

[1] Eyke Hüllermeier, Johannes Fürnkranz, Weiwei Cheng, and Klaus Brinker. Label ranking by

learning pairwise preferences. *Artificial Intelligence*, 172(16):1897–1916, 2008.

[2] Massimo Gurrieri, Philippe Fortemps, and Xavier Siebert. Alternative decomposition techniques for label ranking. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 464–474. Springer, 2014.

[3] Massimo Gurrieri, Xavier Siebert, Philippe Fortemps, Salvatore Greco, and Roman Słowiński. Label ranking: A new rule-based label ranking method. In *Advances on Computational Intelligence*, pages 613–623. Springer, 2012.

[4] Massimo Gurrieri, Philippe Fortemps, and Xavier Siebert. Reduction from label ranking to binary classiffication. In *From Multiple Criteria Decision Aid to Preference Learning.*, pages 3–13. UMons, Université de Mons, Belgium, 2013.

[5] Weiwei Cheng, Jens Hühn, and Eyke Hüllermeier. Decision tree and instance-based learning for label ranking. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 161–168. ACM, 2009.

[6] Johannes Fürnkranz and Eyke Hüllermeier. *Preference learning*. Springer, 2010.

[7] Shankar Vembu and Thomas Gärtner. Label ranking algorithms: A survey. In *Preference learning*, pages 45–64. Springer, 2011.

[8] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[9] Eric E Altendorf, Angelo C Restificar, and Thomas G Dietterich. Learning from sparse data by exploiting monotonicity constraints. *arXiv preprint arXiv:1207.1364*, 2012.

[10] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine learning*, 85(3):333–359, 2011.

[11] Weiwei Cheng, Eyke Hüllermeier, and Krzysztof J Dembczynski. Bayes optimal multilabel classification via probabilistic classifier chains. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 279–286, 2010.

[12] Peter C Fishburn. Binary choice probabilities: on the varieties of stochastic transitivity. *Journal of Mathematical psychology*, 10(4):327–352, 1973.

[13] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques.* MIT press, 2009.

[14] Denis Bouyssou, Thierry Marchant, Marc Pirlot, Alexis Tsoukias, and Philippe Vincke. *Evaluation and decision models with multiple criteria: Stepping stones for the analyst*, volume 86. Springer, 2006.

[15] Noah A Smith and Roy W Tromble. Sampling uniformly from the unit simplex. 2004.

[16] Nabil Aït-Taleb, Brison Valérie, Nicolas Gillis, and Marc Pirlot. Generation of realistic multicriteria datasets using bayesian networks. In *Working Paper*. UMons, Université de Mons, Belgium, 2014.

[17] Nicholas J Higham. Computing the nearest correlation matrix: a problem from finance. *IMA journal of Numerical Analysis*, 22(3):329–343, 2002.

[18] Persi Diaconis and Ronald L Graham. Spearman's footrule as a measure of disarray. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 262–268, 1977.

Session 3

- Invited speaker: "*Boolean functions for classification: logical analysis of data*",
  Yves Crama, University of Liège, Belgium
  Boolean functions are among the simplest and most fundamental objects investigated in mathematics. In spite, or because of their simplicity, they find applications in many scientific fields, including logic, combinatorics, operations research, artificial intelligence, computer science, game theory, engineering, and so forth. In this talk, we present a collection of Boolean models that have been developed over the last 25 years under the name of "Logical Analysis of Data" (or LAD) in order to handle a large variety of classification problems. We focus on the frequent situation where a decision-maker has observed a number of data points (say, vectors of binary attributes) which have been classified either as "positive" or as "negative" examples of a phenomenon under study. The task of the decision-maker is then to develop a classification system that allows her to assign one of the "positive" or "negative" qualifiers to any point that may be presented to her in the future, in a way that remains consistent with the initial observations. We first recall useful facts about partially defined Boolean functions and their extensions, and we introduce the main concepts and definitions used in the LAD framework: support (or "sufficient") sets of attributes, patterns (or "elementary classification rules"), theories (obtained by combining patterns), etc. We show how these building blocks can be used to develop simple interpretable classifiers that perform and generalize well in a variety of experimental situations. Moreover, we argue that these classifiers satisfy some minimal requirements for "justifiability". Finally, we clarify the relation between the LAD classifiers and certain popular classifiers used in the machine learning literature, such as those computed by nearest neighbor classification algorithms or decision trees.

Session 4

- Invited speaker: "*Learning and identifying monotone boolean functions*",
  Endre Boros, Rutgers University, NJ, USA
  Numerous applications require the task of learning and/or identifying a hidden monotone Boolean function.In this talk, first we review several learning models and clarify the the corresponding learning complexity when the hidden function is known to be monotone. The considered models include extending a given partially defined Boolean function or one with missing bits within a specified class of monotone Boolean functions, and learning a certain type of monotone function using membership queries. In the second part of the talk we consider identification problems, which is a special case/extension (depending how one views it) of learning by membership queries. Identification of a monotone function means that we try to generate all of its minimal true (resp. maximal false) points. This problem is strongly related to Boolean dualization or equivalently to finding all minimal transversals of a hypergraph. In this talk we survey some of the related results, and provide a sample of the standard algorithmic techniques.

Session 5

- "*Learning the parameters of a majority rule sorting model taking attribute interactions into account*", Olivier Sobrie[1,2], Vincent Mousseau[1] and Marc Pirlot[2]
  [1] LGI, Ecole Centrale Paris
  [2] MATHRO, Faculté Polytechnique, UMONS

- "*Conjoint axiomatization of the Choquet integral for two-dimensional heterogeneous product sets*",
  Mikhail Timonin, Queen Mary University of London

- "*Utilitaristic Choquistic Regression*",
  Ali Fallah Tehrani[1], Christophe Labreuche[2], Eyke Hullermeier[1]
  [1]Department of Mathematics and Computer Science, University of Marburg,
  [2] Thales Research & Technology

- "*About the french hospitals rankings: a MCDA point of view*",
  Brice Mayag, LAMSADE, Université Paris Dauphine

Learning the parameters of a majority rule sorting model taking attribute interactions into account

Olivier Sobrie[1,3,4] **, Vincent Mousseau**[2] and **Marc Pirlot**[3]

Abstract. We consider a multicriteria sorting procedure based on a majority rule, called MR-Sort. This procedure allows to sort each object of a set, evaluated on multiple criteria, in a category selected among a set of pre-defined and ordered categories. With MR-Sort, the ordered categories are separated by profiles which are vectors of performances on the different attributes. An object is assigned in a category if it is as good as the category lower profile and not better than the category upper profile. To determine if an object is as good as a profile, the weights of the criteria on which the object performances are better than the profile performances are summed up and compared to a threshold. In view of improving the expressiveness of the model, we modify it by introducing capacities to quantify the power of the coalitions. In the paper we describe a mixed integer program and a metaheuristic that give the possibility to learn the parameters of this model from examples of assignment. We test the metaheuristic on real datasets.

1 Introduction

In Multiple Criteria Decision Analysis, the sorting problematic consists in assigning each alternative of a set, evaluated on several monotone criteria, in a category selected among a set of pre-defined and ordered categories. Several MCDA methods are designed to handle such type of problematic. In this paper, we consider a sorting model based on a majority rule, called MR-Sort [11, 17]. In MR-Sort, the categories are separated by profiles which are vectors of performances on the different criteria. Each criterion of the model is associated to a weight representing its importance. Using this model, an alternative is assigned in a category if (a) it is considered at least as good as the category lower profile and (b) it is not considered at least as good as the category upper profile. An alternative is considered as good as a profile if its performances are at least as good as the profile performances on a weighted majority of criteria.

Consider a MR-Sort model composed of 4 criteria (c_1, c_2, c_3 and c_4) and 2 ordered categories ($C_2 \succ C_1$), separated by a profile b_1. Using this model, an alternative is assigned in the "good" category (C_2) iff its performances are as good as the profile b_1 on at least one of the four following minimal criteria coalition:

1. $c_1 \wedge c_2$
2. $c_3 \wedge c_4$
3. $c_1 \wedge c_4$
4. $c_2 \wedge c_4$

A coalition of criteria is said to be minimal if removing any criterion is enough to reject the assertion "alternative a is as good as profile b". Using an additive MR-Sort model, it can be achieved by selecting, for instance, the following weights and majority threshold: $w_1 = 0.3$, $w_2 = 0.2$, $w_3 = 0.1$, $w_4 = 0.4$ and $\lambda = 0.5$. We have $w_1 + w_2 = \lambda$, $w_3 + w_4 = \lambda$, $w_1 + w_4 > \lambda$ and $w_2 + w_4 > \lambda$. All the other coalitions of criteria which are not supersets of the 3 minimal coalitions listed above are not sufficient to be considered as good as b_1 (e.g. $w_1 + w_3 < \lambda$).

Now consider the same type of model, but with the following minimal criteria coalitions:

1. $c_1 \wedge c_2$
2. $c_3 \wedge c_4$

Modeling this classification rule with an additive MR-Sort model is impossible. There exist no weights and majority threshold satisfying solely the 2 minimal criteria coalitions. In view of being able to represent such type of rule, we propose in this paper a new formulation of MR-Sort, called Capacitive-MR-Sort. This formulation expresses the majority rule of MR-Sort with capacities like in the Choquet Integral [8].

The paper is organized as follows. The next section describes formally the MR-Sort model and the new formulation of MR-Sort with capacities. Section 3 recalls the literature dealing with learning parameters of MR-Sort models from assignment examples. The next two sections describe respectively a Mixed Integer Program and a metaheuristic that allow to learn the parameters of a Capacitive-MR-Sort. Some experimental results are finally presented.

2 MR-Sort and Capacitive-MR-Sort

2.1 MR-Sort

MR-Sort is a method for assigning objects in ordered categories. Each object is described by a multicriteria vector of attribute values. The attribute values can be meaningfully ordered, i.e. there is an underlying order on each attribute scale, which is interpreted as a "better than" relation. Categories are determined by limit profiles, which are vectors of attribute values. The lower limit profile of a category is the upper limit profile of the category below. The MR-Sort rule works as follows. An object is assigned to a category if it is better than

[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

the lower limit profile of the category on a sufficiently large coalition of (weighted) attributes and this condition is not met for the upper limit profile of this category. Obviously, MR-Sort is a monotone rule, i.e. an object that is at least as good as another on all attributes cannot be assigned to a lower category.

The MR-Sort rule is a simplified version of the ELECTRE TRI procedure, a method that is used in MCDA to assign objects to predefined categories [19, 16]. The underlying semantic is generally to assign the objects labels such as "good", "average", "bad", ...

Formally, let X be a set of objects evaluated on n ordered attributes (or criteria), $F = \{1, ..., n\}$. We assume that X is the Cartesian product of the criteria scales, $X = \prod_{j=1}^{n} X_j$. An object $a \in X$ is thus a vector $(a_1, \dots, a_j, \dots, a_n)$, where $a_j \in X_j$ for all j.

The ordered categories which the objects are assigned to by the MR-Sort model are denoted by C_h, with $h = 1, \dots, p$. Category C_h is delimited by its lower limit profile b_{h-1} and its upper limit profile b_h, which is also the lower limit profile of category C_{h+1} (provided $0 < h < p$). The profile b_h is the vector of criterion values $(b_{h,1}, \dots, b_{h,j}, \dots, b_{h,n})$, with $b_{h,j} \in X_j$ for all j. We denote by $P = \{1, ..., p-1\}$ the list of profile indices.

By convention, the best category, C_p, is delimited by a fictive upper profile, b_p, and the worst one, C_1, by a fictive lower profile, b_0.

It is assumed that the profiles dominate one another, i.e.:

$$b_{h-1,j} \leq b_{h,j}, \quad h = 1, \dots, p; \; j = 1, \dots, n.$$

Using the MR-Sort procedure, an object is assigned to a category if its criterion values are at least as good as the category lower profile values on a weighted majority of criteria while this condition is not fulfilled when the object's criterion values are compared to the category upper profile values. In the former case, we say that the object is *preferred* to the profile, while, in the latter, it is not. Formally, if an object $a \in X$ is *preferred* to a profile b_h, we denoted this by $a \succcurlyeq b_h$. Object a is preferred to profile b_h whenever the following condition is met:

$$a \succcurlyeq b_h \Leftrightarrow \sum_{j:a_j \geq b_{h,j}} w_j \geq \lambda, \tag{1}$$

where w_j is the nonnegative weight associated with criterion j, for all j and λ sets a majority level. The weights satisfy the normalization condition $\sum_{j \in F} w_j = 1$; λ is called the *majority threshold*; it satisfies $\lambda \in [1/2, 1]$.

The preference relation $\succcurlyeq$ defined by (1) is called an *outranking* relation without veto or a *concordance* relation ([16]; see also [2, 3] for an axiomatic description of such relations).

Consequently, the condition for an object $a \in X$ to be assigned to category C_h writes:

$$\sum_{j:a_j \geq b_{h-1,j}} w_j \geq \lambda \quad \text{and} \quad \sum_{j:a_j \geq b_{h,j}} w_j < \lambda \tag{2}$$

The MR-Sort assignment rule described above involves $pn + 1$ parameters, i.e. n weights, $(p-1)n$ profiles evaluations and one majority threshold. Note that the profiles b_0 and b_p are conventionally defined as follows: $b_{0,j}$ is a value such that $a_j \geq b_{0,j}$ for all $a \in X$ and $j = 1, \dots, n$; $b_{p,j}$ is a value such that $a_j < b_{p,j}$ for all $a \in X$ and $j = 1, \dots, n$.

A *learning set* A is a subset of objects $A \subseteq X$ for which an assignment is known. For $h = 1, \dots, p$, A_h denotes the subset of objects $a \in A$ which are assigned to category C_h. The subsets A_h are disjoint; some of them may be empty.

2.2 Capacitive-MR-Sort

Before describing the Capacitive-MR-Sort model, we introduce the notion of capacity. To illustrate this, we consider an application in which a committee for a higher education program has to decide about the admission of students on basis of their evaluations in 4 courses: math, physics, chemistry and history. To be accepted in the program, the committee judges that a student should have a sufficient majority of evaluations above 10/20. The courses (criteria) coalitions don't have the same importance. The strength of a coalition of courses varies as a function of the courses belonging to the coalition. The committee stated that the following subsets of courses are the minimal coalition of courses in which the evaluation should be above 10/20 in view of being accepted:

- {math, physics}
- {math, chemistry}
- {physics, history}

As an example of this rule, Table 1 shows evaluations of several students and, for each student, if he is accepted or refused.

	Math	Physic	Chemistry	History	A/R
James	11	11	9	9	A
Marc	11	9	11	9	A
Robert	9	9	11	11	A
John	11	9	9	11	R
Paul	9	11	9	11	R
Pierre	9	11	11	9	R

Table 1. Evaluation of students and their acceptance/refusal status

Representing these assignments by using a MR-Sort model with profiles fixed at 10/20 in each course is impossible. There are no weights allowing to model such rules. MR-Sort is not adapted to model such types of rules because it does not handle criteria interactions.

In view of taking criterion interactions into account, we propose to modify the definition of the global outranking relation, $a \succcurlyeq b_h$, given in (1). We introduce the notion of capacity. A capacity is a function $\mu : 2^F \to [0, 1]$ such that:

- $\mu(B) \geq \mu(A)$, for all $A \subseteq B \subseteq F$ (monotonicity) ;
- $\mu(\emptyset) = 0$ and $\mu(F) = 1$ (normalization).

The Möbius transform allows to express the capacity in another form:

$$\mu(A) = \sum_{B \subseteq A} m(B) \tag{3}$$

for all $A \subseteq F$, with $m(B)$ defined as:

$$m(B) = \sum_{C \subseteq B} (-1)^{|B|-|C|} \mu(C) \tag{4}$$

The value $m(B)$ can be interpreted as the weight that is exclusively allocated to B as a whole. A capacity can be defined directly by its Möbius transform also called "interaction". An interaction m is a set function $m : 2^F \rightarrow [-1, 1]$ satisfying the following conditions:

$$\sum_{j \in K \subseteq J \cup \{j\}} m(K) \geq 0 \qquad \forall j \in F, J \subseteq F \backslash \{i\} \tag{5}$$

and

$$\sum_{K \subseteq F} m(K) = 1.$$

If m is an interaction, the set function defined by $\mu(A) = \sum_{B \subseteq A} m(B)$ is a capacity. Conditions (5) guarantee that μ is monotone [5].

Using a capacity to express the weight of the coalition in favor of an object, we transform the outranking rule as follows:

$$a \succcurlyeq b_h \Leftrightarrow \mu(A) \geq \lambda \text{ with } A = \{j : a_j \geq b_{h,j}\} \\ \text{and } \mu(A) = \sum_{B \subseteq A} m(B) \tag{6}$$

Computing the value of $\mu(A)$ with the Möbius transform induces the evaluation of $2^{|A|}$ parameters. In a model composed of n criteria, it implies the elicitation of 2^n parameters, with $\mu(\emptyset) = 0$ and $\mu(F) = 1$. To reduce the number of parameters to elicit, we use a 2-additive capacity in which all the interactions involving more than 2 criteria are equal to zero. In the literature [12], for the ranking problematic, it has been shown experimentally that a 2-additive model allows to improve the representation capabilities. However using a 3-additive capacity instead of a 2-additive one does not significantly improve the accuracy of the model. Inferring a 2-additive capacity for a model having n criteria requires the determination of $\frac{n(n+1)}{2} - 1$ parameters.

Finally, the condition for an object $a \in X$ to be assigned to category C_h can be expressed as follows:

$$\mu(F_{a,h-1}) \geq \lambda \quad \text{and} \quad \mu(F_{a,h}) < \lambda \tag{7}$$

with $F_{a,h-1} = \{j : a_j \geq b_{h-1,j}\}$ and $F_{a,h} = \{j : a_j \geq b_{h,j}\}$.

3 Learning the parameters of a MR-Sort model

Learning the parameters of MR-Sort and ELECTRE TRI models has been already studied in several articles [14, 13, 15, 6, 7, 11, 4, 17, 20]. In this section, we recall how to learn the set of parameters of an MR-Sort using respectively an exact method [11] and a metaheuristic [17].

3.1 Mixed Integer Programming

Learning the parameters of an MR-Sort model using linear programming techniques has been proposed in [11]. The paper describes a Mixed Integer Program (MIP) taking a set of assignment examples and their vector of performances as input and finding the parameters of an MR-Sort model such that a majority of the examples are restored by the inferred model. We recall in this subsection the main steps to obtain the MIP formulation proposed in [11].

The definition of an outranking relation (1) can be rewritten as follows:

$$a \succcurlyeq b_h \iff \sum_{j=1}^{n} c_{a,j}^h \geq \lambda, \text{ with } c_{a,j}^h = \begin{cases} w_j & \text{if } a_j \geq b_{h,j} \\ 0 & \text{otherwise} \end{cases}$$

To linearize this constraint, we introduce for each value $c_{a,j}^h$, a binary variable $\delta_{a,j}^l$ that it is equal to 1 when the performance of the object a is at least equal or better than the performance of the profile b_l on criterion j and 0 otherwise. To obtain the value of $\delta_{a,j}^l$, we add the following constraints:

$$M(\delta_{a,j}^l - 1) \leq a_j - b_{l,j} < M \cdot \delta_{a,j}^l \tag{8}$$

By using the value $\delta_{a,j}^l$, the values of $c_{a,j}^l$ are deduced as follows:

$$\begin{cases} c_{a,j}^l & \leq \delta_{a,j}^l \\ c_{a,j}^l & \leq w_j \\ c_{a,j}^l & \geq \delta_{a,j}^l - 1 + w_j \end{cases}$$

The objective function of the MIP consists in maximizing the number of examples compatible with the learned model, i.e. minimizing the 0/1 loss function. In order to model this, we introduce new binary variables γ_a, equal to 1 if object a is assigned in the expected category, i.e. the category it has been assigned in the learning set, and equal to 0 otherwise. To deduce the value of γ_a variables, two additional constraints are added:

$$\begin{cases} \sum_{j=1}^n c_{a,j}^{h-1} & \geq \lambda + M(\gamma_a - 1) \\ \sum_{j=1}^n c_{a,j}^h & < \lambda - M(\gamma_a - 1) \end{cases}$$

Finally, the combination of all the constraints leads to the MIP given in (9).

3.2 Metaheuristic

The MIP presented in the previous section is not suitable for large datasets because of the high computing time that is required to infer the MR-Sort parameters. In view of learning MR-Sort models in the context of large datasets, a metaheuristic has been proposed in [17]. As in the MIP, the metaheuristic takes as input a set of assignment examples and their vector of performances and returns the parameters of an MR-Sort model.

The metaheuristic proposed in [17] works as follows. First a population of MR-Sort models is initialized. After the initialization, the two following steps are repeated iteratively on each model in the population:

$$\left\{\begin{array}{rcll}
\max & & \displaystyle\sum_{a\in A}\gamma_a & \\
\displaystyle\sum_{j=1}^{n} c_{a,j}^{h-1} & \geq & \lambda + M(\gamma_a - 1) & \forall a \in A_h, h = \{2, ..., p\} \\
\displaystyle\sum_{j=1}^{n} c_{a,j}^{h} & < & \lambda - M(\gamma_a - 1) & \forall a \in A_h, h = \{1, ..., p-1\} \\
a_j - b_{l,j} & < & M \cdot \delta_{a,j}^{l} & \forall j \in F, \forall a \in A_h, \forall h \in P, l = \{h-1, h\} \\
a_j - b_{l,j} & \geq & M(\delta_{a,j}^{l} - 1) & \forall j \in F, \forall a \in A_h, \forall h \in P, l = \{h-1, h\} \\
c_{a,j}^{l} & \leq & \delta_{a,j}^{l} & \forall j \in F, \forall a \in A_h, \forall h \in P, l = \{h-1, h\} \\
c_{a,j}^{l} & \leq & w_j & \forall j \in F, \forall a \in A_h, \forall h \in P, l = \{h-1, h\} \\
c_{a,j}^{l} & \geq & \delta_{a,j}^{l} - 1 + w_j & \forall j \in F, \forall a \in A_h, \forall h \in P, l = \{h-1, h\} \\
b_{h,j} & \geq & b_{h-1,j} & \forall j \in F, h = \{2, ..., p-1\} \\
\displaystyle\sum_{j=1}^{n} w_j & = & 1 & \\
\delta_{a,j}^{l} & \in & \{0, 1\} & \forall j \in F, \forall a \in A_h, \forall h \in P, l = \{h-1, h\} \\
c_{a,j}^{l} & \in & [0, 1] & \forall j \in F, \forall a \in A_h, \forall h \in P, l = \{h-1, h\} \\
b_{h,j} & \in & \mathbb{R} & \forall j \in F, \forall h \in P \\
\gamma_a & \in & \{0, 1\} & \forall a \in X \\
w_j & \in & [0, 1] & \forall j \in F \\
\lambda & \in & [0.5, 1] &
\end{array}\right. \tag{9}$$

1. A linear program optimizes the weights and the majority threshold on basis of assignment examples and fixed profiles.
2. Given the inferred weight and the majority threshold, a heuristic adjusts the profiles of the model on basis of the assignment examples.

After applying these two steps to all the models of the population, the $\lfloor \frac{n}{2} \rfloor$ models restoring the least numbers of examples are reinitialized. These steps are repeated until the metaheuristic finds a model that fully restores all the examples or after a number of iterations given a priori.

The linear program designed to learn the weights and the majority threshold is given by (10). It minimizes a sum of slack variables, x'_a and y'_a, that is equal to 0 when all the objects are correctly assigned, i.e. assigned in the category defined in the input dataset. We remark that the objective function of the linear program does not explicitly minimize the $0/1$ loss but a sum of slacks. It implies that compensatory effects might appears to the detriment of the $0/1$ loss. However in this metaheuristic, we consider that this effects are acceptable. This linear program doesn't contain binary variables, therefore the computing time remains reasonable when the size of the problem increases.

The objective function of the heuristic varying the profiles maximizes the number of examples compatible with the model. To do so, it iterates over each profile h and each criterion j and identifies a set of candidate moves which correspond to the performances of the examples on criterion j located between the profiles $h-1$ and $h+1$. Each candidate move is evaluated as a function of the probability to improve the classification accuracy of the model. To evaluate if a candidate move is likely or unlikely to improve the classification of one or several objects, the examples which have an evaluation on criterion j located between the current value of the profile, $b_{h,j}$ and the candidate move, $b_{h,j} + \delta$ (resp. $b_{h,j} - \delta$) are classified in different subsets:

$V_{h,j}^{+\delta}$ **(resp.** $V_{h,j}^{-\delta}$**)** : the sets of objects misclassified in C_{h+1} instead of C_h (resp. C_h instead of C_{h+1}), for which moving the profile b_h by $+\delta$ (resp. $-\delta$) on j results in a correct assignment.

$W_{h,j}^{+\delta}$ **(resp.** $W_{h,j}^{-\delta}$**)** : the sets of objects misclassified in C_{h+1} instead of C_h (resp. C_h instead of C_{h+1}), for which moving the profile b_h by $+\delta$ (resp. $-\delta$) on j strengthens the criteria coalition in favor of the correct classification but will not by itself result in a correct assignment.

$Q_{h,j}^{+\delta}$ **(resp.** $Q_{h,j}^{-\delta}$**)** : the sets of objects correctly classified in C_{h+1} (resp. C_{h+1}) for which moving the profile b_h by $+\delta$ (resp. $-\delta$) on j results in a misclassification.

$R_{h,j}^{+\delta}$ **(resp.** $R_{h,j}^{-\delta}$**)** : the sets of objects misclassified in C_{h+1} instead of C_h (resp. C_h instead of C_{h+1}), for which moving the profile b_h by $+\delta$ (resp. $-\delta$) on j weakens the criteria coalition in favor of the correct classification but does not induce misclassification by itself.

$T_{h,j}^{+\delta}$ **(resp.** $T_{h,j}^{-\delta}$**)** : the sets of objects misclassified in a category higher than C_h (resp. in a category lower than C_{h+1}) for which the current profile evaluation weakens the criteria coalition in favor of the correct classification.

In order to formally define these sets we introduce the following notation. A_h^l denotes the subset of misclassified objects that are assigned in category C_l by the model while in the dataset, they are assigned in category C_h. $A_{<h}^{>l}$ denotes the subset of misclassified objects that are assigned in category higher than C_l by the model while in the dataset it is assigned in a category below C_h. We denote by $\sigma(a, b_h) =$

$$\left\{\begin{array}{rcll} \min & & \sum_{a \in A} (x'_a + y'_a) & \\ \sum_{j: a_j \geq b_{h-1,j}} w_j - x_a + x'_a & = & \lambda & \forall a \in A_h, \forall h \in P \backslash \{1\} \\ \sum_{j: a_j \geq b_{h,j}} w_j + y_a - y'_a & = & \lambda - \delta & \forall a \in A_h, \forall h \in P \backslash \{p-1\} \\ \sum_{j=1}^{n} w_j & = & 1 & \\ w_j & \in & [0;1] & \forall j \in F \\ \lambda & \in & [0.5;1] & \\ x_a, y_a, x'_a, y'_a & \in & \mathbb{R}_0^+ & \end{array}\right. \tag{10}$$

$\sum_{j:a_j \geq b_{h,j}} w_j$, the sum of criteria weights in favor of object a against profile b_h. We have, for any h, j and positive δ:

$$V_{h,j}^{+\delta} = \left\{a \in A_h^{h+1} : b_{h,j} + \delta > a_j \geq b_{h,j} \text{ and } \sigma(a, b_h) - w_j < \lambda\right\}$$

$$V_{h,j}^{-\delta} = \left\{a \in A_{h+1}^{h} : b_{h,j} - \delta < a_j < b_{h,j} \text{ and } \sigma(a, b_h) + w_j \geq \lambda\right\}$$

$$W_{h,j}^{+\delta} = \left\{a \in A_h^{h+1} : b_{h,j} + \delta > a_j \geq b_{h,j} \text{ and } \sigma(a, b_h) - w_j \geq \lambda\right\}$$

$$W_{h,j}^{-\delta} = \left\{a \in A_{h+1}^{h} : b_{h,j} - \delta < a_j < b_{h,j} \text{ and } \sigma(a, b_h) + w_j < \lambda\right\}$$

$$Q_{h,j}^{+\delta} = \left\{a \in A_{h+1}^{h+1} : b_{h,j} + \delta > a_j \geq b_{h,j} \text{ and } \sigma(a, b_h) - w_j < \lambda\right\}$$

$$Q_{h,j}^{-\delta} = \left\{a \in A_h^{h} : b_{h,j} - \delta < a_j < b_{h,j} \text{ and } \sigma(a, b_h) + w_j \geq \lambda\right\}$$

$$R_{h,j}^{+\delta} = \left\{a \in A_{h+1}^{h} : b_{h,j} + \delta > a_j \geq b_{h,j}\right\}$$

$$R_{h,j}^{-\delta} = \left\{a \in A_h^{h+1} : b_{h,j} - \delta < a_j < b_{h,j}\right\}$$

$$T_{h,j}^{+\delta} = \left\{a \in A_{<h}^{>h} : b_{h,j} + \delta > a_j \geq b_{h,j}\right\}$$

$$T_{h,j}^{-\delta} = \left\{a \in A_{>h+1}^{<h+1} : b_{h,j} - \delta < a_j \leq b_{h,j}\right\}$$

The evaluation of the candidate move is done by aggregating the number of elements in each subset. Finally the choice to move or not the profile on the criterion is determined by comparing the candidate move evaluation to a random number drawn uniformly. These operations are repeated multiple times on each profile and each criterion.

4 Mixed Integer Program to learn a Capacitive-MR-Sort model

As compared to a MR-Sort with additive weights, a MR-Sort model with capacities implies more parameters. In a standard MR-Sort model, a weight is associated to each criterion, which makes overall n parameters to elicit. With an MR-Sort model limited to two-additive capacities, the computation of the weights of a coalition of criteria involves the weights of the criteria in the coalition and the pairwise interactions (Möbius coefficients) between these criteria. Overall there are $n + \frac{n(n-1)}{2} - 1 = \frac{n(n+1)}{2} - 1$ coefficients. In the two-additive case, let us denote by m_j the weights of criterion j and by $m_{j,k}$ the Möbius interactions between criteria j and k. The capacity $\mu(A)$ of a subset of criteria is obtained as: $\mu(A) = \sum_{j \in A} m_j + \sum_{\{j,k\} \subseteq A} m_{j,k}$. The constraints (5) on the interaction read:

$$m_j + \sum_{k \in J} m_{j,k} \geq 0 \qquad \forall j \in F, \forall J \subseteq F \backslash \{j\} \tag{11}$$

and

$$\sum_{j \in F} m_j + \sum_{\{j,k\} \subseteq F} m_{j,k} = 1.$$

The number of monotonicity constraints evolves exponentially as a function of the number of criteria, n. In [10], two other formulations are proposed in order to reduce significantly the number of constraints ensuring the monotonicity of the capacities. The first formulation reduces the number of constraints to $2n^2$ but leads to a non linear program. The second formulation introduces n^2 extra variables and reduces the number of constraints to $n^2 + 1$ without introducing non linearities.

With a 2-additive MR-Sort model, the constraints for an alternative a to be assigned in a category h (7) can also be expressed as follows:

$$\begin{cases} \sum_{j=1}^{n} c_{a,j}^{h-1} + \sum_{j=1}^{n} \sum_{k=1}^{j} c_{a,j,k}^{h-1} & \geq \lambda + M(\gamma_a - 1) \\ \sum_{j=1}^{n} c_{a,j}^{h} + \sum_{j=1}^{n} \sum_{k=1}^{j} c_{a,j,k}^{h} & < \lambda - M(\gamma_a - 1) \end{cases} \tag{12}$$

with:

- $c_{a,j}^{h-1}$ (resp. $c_{a,j}^{h}$) equals m_j if performance of alternative a is at least as good as the performance of profile b_{h-1} (resp. b_h) on criterion j, and equals 0 otherwise;
- $c_{a,j,k}^{h-1}$ (resp. $c_{a,j,k}^{h}$) equals $m_{j,k}$ if performance of alternative a is at least as good as the performance of profile b_{h-1} (resp. b_h) on criteria j and k, and equals 0 otherwise.

For all $a \in X$, $j \in F$ and $l \in P$, constraints (11) imply that $c_{a,j}^{l} \geq 0$ and that $c_{a,j,k}^{l} \in [-1, 1]$. The values of $c_{a,j}^{h-1}$ and $c_{a,j}^{h}$ can be obtained in a similar way as it is done for learning the parameters of a standard MR-Sort model by replacing the weights with the corresponding Möbius coefficient (13).

$$\begin{cases} c_{a,j}^{l} & \leq \delta_{a,j}^{l} \\ c_{a,j}^{l} & \leq m_j \\ c_{a,j}^{l} & \geq \delta_{a,j}^{l} - 1 + m_j \end{cases} \tag{13}$$

However it is not the case for the variables $c_{a,j,k}^{h-1}$ and $c_{a,j,k}^{h}$, because they imply two criteria. To linearize the formulation, we introduce new binary variables, $\Delta^l_{a,j,k}$ equal to 1 if alternative a has better performances than profile b_l on criteria j and k and equal to 0 otherwise. We obtain the value of $\Delta^l_{a,j,k}$ thanks to the conjunction of constraints given at (8) and the following constraints:

$$2\Delta^l_{a,j,k} \leq \delta^l_{a,j} + \delta^k_{a,j} \leq \Delta^l_{a,j,k} + 1$$

In order to deduce the value of $c^l_{a,j,k}$, which can be either positive or negative, for all $l \in P$, we decompose the variable in two parts, $\alpha^l_{a,j,k}$ and $\beta^l_{a,j,k}$ such that $c^l_{a,j,k} = \alpha^l_{a,j,k} - \beta^l_{a,j,k}$ with $\alpha^l_{a,j,k} \geq 0$ and $\beta^l_{a,j,k} \geq 0$. The same is done for $m_{j,k}$ which is decomposed as follows: $m_{j,k} = m^+_{j,k} - m^-_{j,k}$ with $m^+_{j,k} \geq 0$ and $m^-_{j,k} \geq 0$. The value of $\alpha^l_{a,j,k}$ and $\beta^l_{a,j,k}$ are finally obtained thanks to the following constraints:

$$\begin{cases} \alpha^l_{a,j,k} & \leq \Delta^l_{a,j,k} \\ \alpha^l_{a,j,k} & \leq m^+_{j,k} \\ \alpha^l_{a,j,k} & \geq \Delta^l_{a,j,k} - 1 + m^+_{j,k} \end{cases} \qquad \begin{cases} \beta^l_{a,j,k} & \leq \Delta^l_{a,j,k} \\ \beta^l_{a,j,k} & \leq m^-_{j,k} \\ \beta^l_{a,j,k} & \geq \Delta^l_{a,j,k} - 1 + m^-_{j,k} \end{cases}$$

Finally, we obtain the MIP given in (14).

5 Metaheuristic to learn a Capacitive-MR-Sort model

The MIP described in the previous section requires a lot of binary variables and is therefore unsuitable for large problems. In subsection 3.2, we described the principle of a metaheuristic designed to learn the parameters of an MR-Sort model. In this section, we describe an adaptation of the metaheuristic in view of learning the parameters of a Capacitive-MR-Sort model. Like for the MIP described in the previous section, we limit the model to 2-additive capacities in order to reduce the number of coefficient in comparison to a model with a general capacity.

The main component that needs to be adapted in the metaheuristic in order to be able to learn a Capacitive-MR-Sort model is the linear program that infers the weights and the majority threshold (10). Like in the MIP described in the previous section, we use the Möbius transform to express capacities. In view of inferring Möbius coefficients, m_j and $m_{j,k}$, $\forall j, \forall k$ with $k < j$, we modify the linear program as given in (15).

The value of $x_a - x'_a$ (resp. $y_a - y'_a$) represents the difference between the capacity of the criteria belonging to the coalition in favor of $a \in A_h$ w.r.t. b_{h-1} (resp. b_h) and the majority threshold. If both $x_a - x'_a$ and $y_a - y'_a$ are positive, then the object a is assigned to the right category. In order to try to maximize the number of examples correctly assigned by the model, the objective function of the linear program minimizes the sum of x'_a and y'_a, i.e. the objective function is $\min \sum_{a \in A}(x'_a + y'_a)$.

The heuristic adjusting the profile also needs some adaptations in view of taking capacities into account. More precisely, it is needed to adapt the formal definition of the sets in which objects are classified for computing the candidate move evaluation. The semantic of the sets, described in Section 3.2 remains the same, only the formal definitions of the sets are adapted as follows.

$$V^{+\delta}_{h,j} = \left\{a \in A^{h+1}_h : b_{h,j} + \delta > a_j \geq b_{h,j} \text{ and } \mu(F_{a,h} \backslash \{j\}) < \lambda\right\}$$
$$V^{-\delta}_{h,j} = \left\{a \in A^h_{h+1} : b_{h,j} - \delta < a_j < b_{h,j} \text{ and } \mu(F_{a,h} \cup \{j\}) \geq \lambda\right\}$$
$$W^{+\delta}_{h,j} = \left\{a \in A^{h+1}_h : b_{h,j} + \delta > a_j \geq b_{h,j} \text{ and } \mu(F_{a,h} \backslash \{j\}) \geq \lambda\right\}$$
$$W^{-\delta}_{h,j} = \left\{a \in A^h_{h+1} : b_{h,j} - \delta < a_j < b_{h,j} \text{ and } \mu(F_{a,h} \cup \{j\}) < \lambda\right\}$$
$$Q^{+\delta}_{h,j} = \left\{a \in A^{h+1}_{h+1} : b_{h,j} + \delta > a_j \geq b_{h,j} \text{ and } \mu(F_{a,h} \backslash \{j\}) < \lambda\right\}$$
$$Q^{-\delta}_{h,j} = \left\{a \in A^h_h : b_{h,j} - \delta < a_j < b_{h,j} \text{ and } \mu(F_{a,h} \cup \{j\}) \geq \lambda\right\}$$

The formal definitions of the sets $R^{+\delta}_{h,j}$, $R^{-\delta}_{h,j}$, $T^{+\delta}_{h,j}$ remain the same as for the simple additive MR-Sort model as well as function computing the evaluations taking into account the size of the sets.

6 Experimentations

The use of the MIP for learning a Capacitive-MR-Sort model is limited because of the high number of binary variables it involves. It contains more binary variables than the MIP learning the parameters of a simple additive MR-Sort model. In [11], experiments have demonstrated that the computing time required to learn the parameters of a standard MR-Sort model having a small number of criteria and categories from a small set of assignment examples becomes quickly prohibitive. Therefore we cannot expect to be able to treat large problems using the MIP learning Capacitive-MR-Sort models.

In view of assessing the performances of the metaheuristic designed for learning the parameters of a Capacitive-MR-Sort model, we used it to learn Capacitive-MR-Sort models from several real datasets presented in Table 2. These datasets have been found in the UCI machine learning repository [1] and in WEKA [9]. They have been already used to assess the learning performances of other algorithms, like in [18] and [17]. The dataset presented in Table 2 contains from 120 to 1728 instances, 4 to 8 criteria (criteria) and 2 to 36 categories. In the experimentations, the categories have been binarized by thresholding at the median (like in [18, 17]). All the input criteria of the datasets are considered as monotone.

Dataset	#instances	#criteria	#categories
DBS	120	8	2
CPU	209	6	4
BCC	286	7	2
MPG	392	7	36
ESL	488	4	9
MMG	961	5	2
ERA	1000	4	9
LEV	1000	4	5
CEV	1728	6	4

Table 2. Datasets

In a first experiment, we used 50% of the alternatives contained in the datasets as learning set and the rest as test

$$\left\{\begin{array}{rcll}
\max & & \displaystyle\sum_{a\in A}\gamma_a & \\
\displaystyle\sum_{j=1}^{n} c_{a,j}^{h-1}+\sum_{j=1}^{n}\sum_{k=1}^{j}\alpha_{a,j,k}^{h-1}-\sum_{j=1}^{n}\sum_{k=1}^{j}\beta_{a,j,k}^{h-1} & \geq & \lambda+M(\gamma_a-1) & \forall a\in A_h, h=2,...,p\\
\displaystyle\sum_{j=1}^{n} c_{a,j}^{h}+\sum_{j=1}^{n}\sum_{k=1}^{j}\alpha_{a,j,k}^{h}-\sum_{j=1}^{n}\sum_{k=1}^{j}\beta_{a,j,k}^{h} & < & \lambda-M(\gamma_a-1) & \forall a\in A_h, \forall h\in P\\
c_{a,j}^{l} & \leq & \delta_{a,j}^{l} & \forall j\in F, \forall a\in A_h, \forall h\in P, l=\{h-1,h\}\\
c_{a,j}^{l} & \leq & m_j & \forall j\in F, \forall a\in A_h, \forall h\in P, l=\{h-1,h\}\\
c_{a,j}^{l} & \geq & \delta_{a,j}^{l}-1+m_j & \forall j\in F, \forall a\in A_h, \forall h\in P, l=\{h-1,h\}\\
a_j-b_{l,j} & < & M\cdot\delta_{a,j}^{l} & \forall j\in F, \forall a\in A_h, \forall h\in P, l=\{h-1,h\}\\
a_j-b_{l,j} & \geq & M(\delta_{a,j}^{l}-1) & \forall j\in F, \forall a\in A_h, \forall h\in P, l=\{h-1,h\}\\
\delta_{a,j}^{l}+\delta_{a,k}^{l} & \geq & 2\Delta_{a,j,k}^{l} & \forall j\in F, \forall k\in F, k<j, \forall a\in A_h, \forall h\in P, l=\{h-1,h\}\\
\delta_{a,j}^{l}+\delta_{a,k}^{l} & \leq & \Delta_{a,j,k}^{l}+1 & \forall j\in F, \forall k\in F, k<j, \forall a\in A_h, \forall h\in P, l=\{h-1,h\}\\
\alpha_{a,j,k}^{l} & \leq & \Delta_{a,j,k}^{l} & \forall j\in F, \forall k\in F, k<j, \forall a\in A_h, \forall h\in P, l=\{h-1,h\}\\
\alpha_{a,j,k}^{l} & \leq & m_{j,k}^{+} & \forall j\in F, \forall k\in F, k<j, \forall a\in A_h, \forall h\in P, l=\{h-1,h\}\\
\alpha_{a,j,k}^{l} & \geq & \Delta_{a,j,k}^{l}-1+m_{j,k}^{+} & \forall j\in F, \forall k\in F, k<j, \forall a\in A_h, \forall h\in P, l=\{h-1,h\}\\
\beta_{a,j,k}^{l} & \leq & \Delta_{a,j,k}^{l} & \forall j\in F, \forall k\in F, k<j, \forall a\in A_h, \forall h\in P, l=\{h-1,h\}\\
\beta_{a,j,k}^{l} & \leq & m_{j,k}^{-} & \forall j\in F, \forall k\in F, k<j, \forall a\in A_h, \forall h\in P, l=\{h-1,h\}\\
\beta_{a,j,k}^{l} & \geq & \Delta_{a,j,k}^{l}-1+m_{j,k}^{-} & \forall j\in F, \forall k\in F, k<j, \forall a\in A_h, \forall h\in P, l=\{h-1,h\}\\
m_j+\displaystyle\sum_{k\in J}(m_{j,k}^{+}-m_{j,k}^{-}) & \geq & 0 & \forall j\in F, \forall J\subseteq F\backslash\{j\}\\
b_{h,j} & \geq & b_{h-1,j} & \forall j\in F, h=\{2,...,p-1\}\\
\displaystyle\sum_{j=1}^{n} m_j+\sum_{j=1}^{n}\sum_{k=1}^{j}(m_{j,k}^{+}-m_{j,k}^{-}) & = & 1 & \\
c_{a,j}^{l} & \in & [0,1] & \forall j\in F, \forall a\in A_h, \forall h\in P, l=\{h-1,h\}\\
\delta_{a,j}^{l} & \in & \{0,1\} & \forall j\in F, \forall a\in A_h, \forall h\in P, l=\{h-1,h\}\\
\alpha_{a,j,k}^{l}, \beta_{a,j,k}^{l} & \in & [0,1] & \forall j\in F, \forall k\in F, k<j, \forall a\in A_h, \forall h\in P, l=\{h-1,h\}\\
\Delta_{a,j,k}^{l} & \in & \{0,1\} & \forall j\in F, \forall k\in F, k<j, \forall a\in A_h, \forall h\in P, l=\{h-1,h\}\\
m_j & \in & [0,1] & \forall j\in F\\
m_{j,k}^{+}, m_{j,k}^{-} & \in & [0,1] & \forall j\in F, \forall k\in F, k<j\\
b_{h,j} & \in & \mathbb{R} & \forall j\in F, \forall h\in P\\
\gamma_a & \in & \{0,1\} & \forall a\in X\\
\lambda & \in & [0,1] &
\end{array}\right. \tag{14}$$

set. From the examples of the learning set, we learned MR-Sort and Capacitive-MR-Sort models with the metaheuristic. We repeated the operation for 100 random splittings of the datasets in learning and test sets. The results are given in Table 3. We see that the average classification accuracy obtained with the Capacitive-MR-Sort metaheuristic is in average comparable to the one obtained with the MR-Sort metaheuristic. For some datasets, the Capacitive-MR-Sort metaheuristic gives better results but sometimes it is the contrary. The use of a more descriptive model does not help to improve the classification accuracy of the test set.

The second experiment we did consisted in using all the instances of the datasets as learning set. As in the first experiment, for each dataset, we run the two metaheuristic with 100 different seeds. The average classification accuracy and the standard deviation of the learning set of each dataset is

Dataset	META MR-Sort	META Capa-MR-Sort
DBS	0.8400 ± 0.0456	0.8306 ± 0.0466
CPU	0.9270 ± 0.0294	0.9203 ± 0.0315
BCC	0.7271 ± 0.0379	0.7262 ± 0.0377
MPG	0.8174 ± 0.0290	0.8167 ± 0.0468
ESL	0.8992 ± 0.0195	0.9018 ± 0.0172
MMG	0.8303 ± 0.0154	0.8318 ± 0.0121
ERA	0.6905 ± 0.0192	0.6927 ± 0.0165
LEV	0.8454 ± 0.0221	0.8445 ± 0.0223
CEV	0.9217 ± 0.0067	0.9187 ± 0.0153

Table 3. Average and standard deviation of the classification accuracy of the test set when 50 % of examples used as learning set and the rest as test set

$$\left\{\begin{array}{rcll} \min & & \sum_{a\in A}(x'_a+y'_a) & \\ \sum_{j:a_j\geq b_{h-1,j}}^{n}\left(m_j+\sum_{k:a_k\geq b_{h-1,k}}^{j} m_{j,k}\right)-x_a+x'_a & = & \lambda & \forall a\in A_h, \forall h\in P\backslash\{1\} \\ \sum_{j:a_j\geq b_{h,j}}^{n}\left(m_j+\sum_{k:a_k\geq b_{h,k}}^{j} m_{j,k}\right)+y_a-y'_a & = & \lambda-\varepsilon & \forall a\in A_h, \forall h\in P\backslash\{p-1\} \\ \sum_{j=1}^{n} m_j+\sum_{j=1}^{n}\sum_{k=1}^{j} m_{j,k} & = & 1 & \\ m_j+\sum_{k\in J} m_{j,k} & \geq & 0 & \forall j\in F, \forall J\subseteq F\backslash\{j\} \\ \lambda & \in & [0.5;1] & \\ m_j & \in & [0,1] & \forall j\in F \\ m_{j,k} & \in & [-1,1] & \forall j\in F, \forall k\in F, k<j \\ x_a, y_a, x'_a, y'_a & \in & \mathbb{R}_0^+ & a\in A. \end{array}\right. \tag{15}$$

given in Table 4. The Capacitive-MR-Sort metaheuristic does not always give better results than the MR-Sort one.

Dataset	META MR-Sort	META Capa-MR-Sort
DBS	0.9318 ± 0.0036	0.9247 ± 0.0099
CPU	0.9761 ± 0.0000	0.9694 ± 0.0072
BCC	0.7737 ± 0.0013	0.7700 ± 0.0077
MPG	0.8418 ± 0.0000	0.8418 ± 0.0000
ESL	0.9180 ± 0.0000	0.9180 ± 0.0000
MMG	0.8491 ± 0.0011	0.8508 ± 0.0005
ERA	0.7142 ± 0.0028	0.7158 ± 0.0004
LEV	0.8650 ± 0.0000	0.8650 ± 0.0000
CEV	0.9225 ± 0.0000	0.9225 ± 0.0000

Table 4. Average and standard deviation of the classification accuracy of the learning set when using the MR-Sort and Capacitive-MR-Sort models when using all the dataset as learning set

The average computing time required to obtain the results presented in Table 4 is given in Table 5. We observe that learning a Capacitive-MR-Sort model can take up to almost 3 times the time required to learn the parameters of a simple MR-Sort model.

Dataset	META MR-Sort	META Capa-MR-Sort
DBS	3.0508	6.9547
CPU	3.1646	5.2069
BCC	3.3700	7.7545
MPG	4.4136	9.9294
ESL	3.8466	7.2495
MMG	6.1481	13.4848
ERA	5.9689	14.4875
LEV	5.8986	13.2356
CEV	11.1122	31.7042

Table 5. Average computing time (in seconds) required to find a solution with MR-Sort and Capacitive-MR-Sort metaheuristic when using all the examples as learning set

The two experiments show that using a more expressive model does not always result in a better classification accuracy. This observation raises two questions. Firstly, in view of the results obtained, one may doubt that the Capacitive-MR-Sort extends much the original MR-Sort. For what type of assignment data is the new model more flexible? Secondly, is the metaheuristic well-adapted to learn Capacitive-MR-Sort models? To answer these questions, more experimentations have to be done.

7 Comments

We observe that using 2-additive weights instead of simple additive weights in MR-Sort does not result in significant improvement of the 0/1 loss. It is somewhat surprising because the model is more flexible when 2-additive weights are used.

In view of understanding better how the representation capabilities of an MR-Sort model can be improved by using 2-additive weights, we do the following experimentation. We modify the MIP presented in section 3.1 to learn only the weights and the majority threshold of an MR-Sort model on basis of fixed profiles and assignment examples. The objective function of the MIP remains the minimization of the 0/1 loss. The MIP is used to learn the parameters of an MR-Sort model composed of 2 categories, $C_1 \succ C_2$, 4 to 6 criteria, and a fixed profile equals to 0.5 on all the criteria. Each of this learning sets contains 2^n alternatives, with n being the number of criteria of the model that is learnt. Performances of the alternatives of the learning are either equal to 0 or 1 on each criterion and the learning set is built such that each vector of performances is represented once and only once. Alternatives in the learning set are assigned either in C_1 or C_2 such that monotonicity is guaranteed in assignments, i.e. an alternative, x, which has at least equal or better performances on each criterion than another one, y, is never assigned in a least preferred category than the category in which y is assigned. In the experiment, we consider all the non-additive learning sets, i.e. all the learning sets which are not fully compatible with a simple additive MR-Sort model composed of n criteria.

Results of the experimentation are presented in Table 7.

Each row of the table contains the results for a given number of criteria, n. The second column contains the percentage of learning sets that are not compatible with a simple additive MR-Sort model composed of n criteria, among all the learning sets combinations. The last three columns contain the min, max and average percentage of 2^n examples that cannot be restored by a simple additive model among the non-additive learning sets. We observe that a MR-Sort model composed of 4 criteria is, in worst case, not able to restore 6.2% of the examples of the learning set (1 example on 16). With 5 and 6 criteria, the maximum 0/1 loss increases respectively to 9.4% and 12.4%. We see that the proportion of the alternatives that cannot be restored with a simple MR-Sort model is small. This observation might explain the poor gain observed with the Capacitive-MR-Sort metaheuristic compared to the MR-Sort one.

n	% non-additive	MR-Sort		
		min.	max.	avg.
4	11 %	6.2 %	6.2 %	6.2 %
5	57 %	3.1 %	9.4 %	3.9 %
6	97 %	1.6 %	12.5 %	4.8 %

Table 6. Average, minimum and maximum 0/1 loss of the learning sets after learning additive weights and the majority threshold of an MR-Sort model

8 Conclusion

In this paper, we proposed an extension of the MR-Sort model by adding capacitive weights to the model. We called it Capacitive-MR-Sort. We also modified the MIP presented in [11] and the metaheuristic described in [17] in view of being able to learn Capacitive-MR-Sort models. The MIP formulation induces a lot of binary variables and is unsuitable for problems involving large datasets. As we want to be able to deal with real datasets, which are often large, we made experiments with the metaheuristic. Tests have been done on well-known datasets and showed that a more flexible model, the Capacitive-MR-Sort, does not guarantee to get a better classification accuracy. More experiments have to be done in view of being able to better measure and compare the representation ability of MR-Sort and Capacitive-MR-Sort models.

REFERENCES

[1] Bache, K., Lichman, M.: UCI machine learning repository (2013), http://archive.ics.uci.edu/ml

[2] Bouyssou, D., Pirlot, M.: A characterization of concordance relations. European Journal of Operational Research 167(2), 427–443 (2005)

[3] Bouyssou, D., Pirlot, M.: Further results on concordance relations. European Journal of Operational Research 181, 505–514 (2007)

[4] Cailloux, O., Meyer, P., Mousseau, V.: Eliciting ELECTRE TRI category limits for a group of decision makers. European Journal of Operational Research 223(1), 133–140 (2012)

[5] Chateauneuf, A., Jaffray, J.: Derivation of some results on monotone capacities by Möbius inversion. In: Bouchon-Meunier, B., Yager, R.R. (eds.) Uncertainty in Knowledge-Based Systems, International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU '86, Paris, France, June 30 - July 4, 1986, Selected and Extended Contributions. Lecture Notes in Computer Science, vol. 286, pp. 95–102. Springer (1986), http://dx.doi.org/10.1007/3-540-18579-8_8

[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] Grabisch, M.: The application of fuzzy integrals in multicriteria decision making. European Journal of Operational Research 89(3), 445 – 456 (1996), http://www.sciencedirect.com/science/article/pii/037722179500176X

[9] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: An update. SIGKDD Exploration Newsletter 11(1), 10–18 (Nov 2009), http://doi.acm.org/10.1145/1656274.1656278

[10] Hüllermeier, E., Tehrani, A.: Efficient learning of classifiers based on the 2-additive Choquet integral. In: Moewes, C., Nürnberger, A. (eds.) Computational Intelligence in Intelligent Data Analysis, Studies in Computational Intelligence, vol. 445, pp. 17–29. Springer Berlin Heidelberg (2013), http://dx.doi.org/10.1007/978-3-642-32378-2_2

[11] 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)

[12] Meyer, P., Pirlot, M.: On the expressiveness of the additive value function and the choquet integral models. In: DA2PL 2012 Workshop From Multiple Criteria Decision Aid to Preference Learning. pp. 48–56 (2012)

[13] 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)

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

[15] 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)

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

[17] Sobrie, O., Mousseau, V., Pirlot, M.: Learning a majority rule model from large sets of assignment examples. In: Perny, P., Pirlot, M., Tsoukiás, A. (eds.) Algorithmic Decision Theory. pp. 336–350. Springer (2013)

[18] Tehrani, A.F., Cheng, W., Dembczynski, K., Hüllermeier, E.: Learning monotone nonlinear models using the Choquet integral. Machine Learning 89(1-2), 183–211 (2012)

[19] 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)

[20] Zheng, J., Metchebon, S., Mousseau, V., Pirlot, M.: Learning criteria weights of an optimistic Electre Tri sorting rule. Computers & OR 49(0), 28 – 40 (2014), http://www.sciencedirect.com/science/article/pii/S0305054814000677

Conjoint axiomatization of the Choquet integral for two-dimensional heterogeneous product sets

Mikhail Timonin[1]

Abstract. We propose an axiomatization of the Choquet integral model for the general case of a heterogeneous product set $X = X_1 \times X_2$. Previous axiomatizations of the Choquet integral have been given for particular cases $X = Y^n$ and $X = \mathbb{R}^n$. The major difference of this paper from the earlier axiomatizations is that the notion of "comonotonicity" cannot be used in the heterogeneous structure as there does not exist a "built-in" order between elements of sets X_1 and X_2. However, such an order is implied by the representation. Our characterization does not assume commensurateness of criteria a priori. We construct the representation and study its uniqueness properties.

1 Introduction

We propose a representation theorem for the Choquet integral model. Binary relation $\succcurlyeq$ is defined on a heterogeneous product set $X = X_1 \times X_2$. In multicriteria decision analysis (MCDA), elements of the set X are interpreted as alternatives characterized by two criteria taking values from sets X_1 and X_2. Previous axiomatizations of the Choquet integral model have been given for the special cases of $X = Y^n$ (see [Köbberling and Wakker, 2003] for a review of approaches) and $X = \mathbb{R}^n$ (see [Grabisch and Labreuche, 2008] for a review). One related result is the recent axiomatization of the Sugeno integral model ([Greco et al., 2004, Bouyssou et al., 2009]). Another approach using conditions on the utility functions was proposed in [Labreuche, 2012]. The "conjoint" axiomatization of the Choquet integral for the case of a general X was an open problem in the literature [Bouyssou et al., 2012]. The crucial difference with the previous axiomatizations is that the notion of "comonotonicity" cannot be used in the heterogeneous case, due to the fact that there does not exist a meaningful "built-in" order between elements of sets X_1 and X_2. New axioms and modifications of proof techniques had to be introduced to account for that.

Our axioms aim to reflect the main properties of the Choquet integral. The first one is that the set X can be partitioned into subsets, such that within every such subset the preference relation can be represented by an additive function. The axiom (**A3**) we introduce is similar to the "2-graded" condition previously used for characterizing of MIN/MAX and the Sugeno integral ([Greco et al., 2004, Bouyssou et al., 2009]). At every point $z \in X$ it is possible to build two "rectangular cones": $\{x : x_1p \succcurlyeq z_1p, az_2 \succcurlyeq ax_2\}$ for all $p \in X_2$ and all $a \in X_1$, and $\{x : ax_2 \succcurlyeq az_2, z_1p \succcurlyeq x_1p\}$ for all $p \in X_2$ and all $a \in X_1$. The axiom states that triple cancellation must then hold on at least one of these cones. The second property is that the additive representations on different subsets are interrelated, in particular "trade-offs" between criteria values are "consistent" across partition elements both within the same dimension and across different ones. This is reflected by two axioms (**A4, A5**), similar to the ones used in [Wakker, 1991a] and [Krantz et al., 1971] (section 8.2). One, roughly speaking, states that triple cancellation holds across cones, while the other says that ordering of intervals on any dimension must be preserved when they are "projected" onto another dimension by means of equivalence relations. These axioms are complemented by a new condition called bi-independence (**A6**), weak separability (**A2**) [Bouyssou et al., 2009] - which together reflect the monotonicity property of the integral, and the standard essentiality, "comonotonic" Archimedean axiom and restricted solvability (**A7,A8,A9**). Finally, $\succcurlyeq$ is supposed to be a weak order, and X is order dense (**A1**).

2 Choquet integral in MCDA

Definition 1. *Let $N = \{1, \dots, n\}$ be a finite set and 2^N its power set. Capacity (non-additive measure, fuzzy measure) is a set function $\nu : 2^N \to \mathbb{R}_+$ such that:*

1. *$\nu(\varnothing) = 0$;*
2. *$A \subseteq B \Rightarrow \nu(A) \leq \nu(B),\ \forall A, B \in 2^N$.*

In this paper, it is also assumed that capacities are normalized, i.e. $\nu(N) = 1$.

Definition 2. *The Choquet integral with respect to a capacity ν of a function $f : N \to \mathbb{R}$ with values $\{f_1, \dots, f_n\}$ is defined as:*

$$C(\nu, (f_1, \dots, f_n)) = \sum_{i=1}^{n} (f_{(i)} - f_{(i-1)}) \nu(\{j \in N \colon f_j \geq f_{(i)}\})$$

where $f_{(1)}, \dots, f_{(n)}$ is a permutation of $f_1, \dots, f_n$ such that $f_{(1)} \leq f_{(2)} \leq \dots \leq f_{(n)}$, and $f_{(0)} = 0$.

2.1 The model

Let $\succcurlyeq$ be a binary relation on the set $X = X_1 \times X_2$. $\succ, \prec, \preccurlyeq, \sim, \not\succ$ are defined in the usual way. In MCDA, elements of set X are interpreted as alternatives characterized by criteria from the set $N = \{1, 2\}$. Sets X_1 and X_2 contain criteria values for criteria 1 and 2 respectively. We say that $\succcurlyeq$ can be represented by a Choquet integral, if there exists a capacity ν and functions $f_1 : X_1 \to \mathbb{R}$ and $f_2 : X_2 \to \mathbb{R}$, called value functions, such that:

$$x \succcurlyeq y \iff C(\nu, (f_1(x_1), f_2(x_2)) \geq C(\nu, (f_1(y_1), f_2(y_2)).$$

As seen in the definition of the Choquet integral, its calculation involves comparison of f_i's to each other. It is not

[1] Queen Mary University of London, email: m.timonin@qmul.ac.uk

obvious how this operation can be performed in a sensible way in the case of a heterogeneous X. It is well known that direct comparison of value functions for various attributes is not meaningful in the additive model [Krantz et al., 1971], since the origin of each value function can be changed independently. In the homogeneous case $X = Y^n$ this problem is readily solved, as we have a single set Y ("consequences" in the context of decision making under uncertainty). The required order on Y is either assumed as given [Wakker, 1991b] or is readily derived from the ordering of "constant" acts $(y, \dots, y)$ [Wakker, 1991a]. Since there is only one ("consequence") set, we also only have one value function $U : Y \to \mathbb{R}$, and thus comparing $U(y_1)$ to $U(y_2)$ is perfectly valid, since U represents the order on the set Y. None of these methods can be easily transferred to the heterogeneous case.

2.2 Properties of the Choquet integral

Given below are some important properties of the Choquet integral:

1. Functions $f : N \to \mathbb{R}$ and $g : N \to \mathbb{R}$ are comonotonic if for no $i, j \in N$ holds $f(i) > f(j)$ and $g(i) < g(j)$. For all comonotonic f the Choquet integral reduces to a usual Lebesgue integral. In the finite case, the integral is accordingly reduced to a weighted sum.
2. Particular cases of the Choquet integral (e.g. [Grabisch and Labreuche, 2008]). Assume $N = \{1, 2\}$.
   - If $\nu(\{1\}) = \nu(\{2\}) = 1$, then $C(\nu, (f_1, f_2)) = \max(f_1, f_2)$.
   - If $\nu(\{1\}) = \nu(\{2\}) = 0$, then $C(\nu, (f_1, f_2)) = \min(f_1, f_2)$.
   - If $\nu(\{1\}) + \nu(\{2\}) = 1$, then $C(\nu, (f_1, f_2)) = \nu(\{1\})f_1 + \nu(\{2\})f_2$

Property 1 states that the set X can be partitioned into subsets corresponding to particular ordering of the value functions. In the case of two criteria there are only two such sets: $\{x \in X : f_1(x_1) \geq f_2(x_2)\}$ and $\{x \in X : f_2(x_2) \geq f_1(x_1)\}$. Since the integral on each of the sets is reduced to a weighted sum, i.e. $\succcurlyeq$ has an additive representation, we should expect many of the axioms of the additive conjoint model to be valid on this subsets. This is the intuition behind several of the axioms given in the following section.

3 Axioms

Definition 3. *A relation $\succcurlyeq$ on $X_1 \times X_2$ satisfies triple cancellation provided that, for every $a, b, c, d \in X_1$ and $p, q, r, s \in X_2$, if $ap \preccurlyeq bq$, $ar \succcurlyeq bs$, and $cp \succcurlyeq dq$, then $cr \succcurlyeq ds$.*

Definition 4. *A relation $\succcurlyeq$ on $X_1 \times X_2$ is independent iff, for $a, b \in X_1$, $ap \succcurlyeq bp$ for some $p \in X_2$ implies that $aq \succcurlyeq bq$ for every $q \in X_2$; and, for $p, q \in X_2$, $ap \succcurlyeq aq$ for some $a \in X_1$ implies that $bp \succcurlyeq bq$ for every $b \in X_1$.*

A1. $\succcurlyeq$ is a weak order.

A2. Weak separability - for any $a_ip_j, b_ip_j \in X$ such that $a_ip_j \succ b_ip_j$, it holds $a_iq_j \succcurlyeq b_iq_j$ for all $q_j \in X_j$, for $i, j \in \{1, 2\}$.

Note, that from this follows, that for any $a, b \in X_1$ either $ap \succcurlyeq bp$ or $bp \succcurlyeq ap$ for all $p \in X_2$ (symmetrically for the second coordinate). This allows to introduce the following definitions:

Definition 5. *For all $a, b \in X_1$ define $\succcurlyeq_1$ as $a \succcurlyeq_1 b \iff ap \succcurlyeq bp$ for all $p \in X_2$. Define $\succcurlyeq_2$ symmetrically.*

Definition 6. *We call $a \in X_1$ "minimal" if $b \succcurlyeq_1 a$ for all $b \in X_1$, and "maximal" if $a \succcurlyeq_1 b$ for all $b \in X_1$. Symmetric definitions hold for X_2.*

Definition 7. *For any $z \in X$ define $\mathbf{SE}^z = \{x\colon x \in X, x_1 \succcurlyeq_1 z_1, z_2 \succcurlyeq_2 x_2\}$, and $\mathbf{NW}^z = \{x\colon x \in X, x_2 \succcurlyeq_2 z_2, z_1 \succcurlyeq_1 x_1\}$.*

The "rectangular"cones $\mathbf{SE}^z$ and $\mathbf{NW}^z$ play a significant role in the sequel.

A3. For any $z \in X$, triple cancellation holds either for on $\mathbf{SE}^z$ or on $\mathbf{NW}^z$.

A slightly modified version of this axiom holds in the n-dimensional case as well. The axiom says that the set X can be covered by "rectangular" cones, such that triple cancellation holds within each cone. We will call such cones "3C-cones". The axiom effectively partitions X into subsets, defined as follows.

Definition 8. *We say that*

- *$x \in \mathbf{SE}$ if:*
  - *There exists $z \in X$ such that z_1 is not maximal and z_2 is not minimal, triple cancellation holds on $\mathbf{SE}^z$, and $x \in \mathbf{SE}^z$, or*
  - *x_1 is maximal or x_2 is minimal and for no $y \in \mathbf{SE}^x \setminus x$ triple cancellation holds on $\mathbf{NW}^x$;*
- *$x \in \mathbf{NW}$ if:*
  - *There exists $z \in X$ such that z_1 is not maximal and z_2 is not minimal, triple cancellation holds on $\mathbf{SE}^z$, and $x \in \mathbf{SE}^z$, or*
  - *x_1 is minimal or x_2 is maximal and for no $y \in \mathbf{NW}^x \setminus x$ triple cancellation holds on $\mathbf{SE}^x$.*

Define also $\Theta = \{x : x \in \mathbf{NW}, x \in \mathbf{SE}\}$.

Definition 9. *We say that $i \in N$ is essential on $A \subset X$ if there exist $x_ix_j, y_ix_j \in A$, $i, j \in N$, such that $x_ix_j \succ y_ix_j$.*

A4. Whenever $ap \preccurlyeq bq$, $ar \succcurlyeq bs$, $cp \succcurlyeq dq$, it holds that $cr \succcurlyeq ds$, provided that either:

a) $ap, bq, ar, bs, cp, dq, cr, ds \in \mathbf{NW}(\mathbf{SE})$, or;

b) $ap, bq, ar, bs \in \mathbf{NW}$ and $i = 2$ is essential on $\mathbf{NW}$ and $cp, dq, cr, ds \in \mathbf{SE}$ or vice versa, or;

c) $ap, bq, cp, dq \in \mathbf{NW}$ and $i = 1$ is essential on $\mathbf{NW}$ and $cp, dq, cr, ds \in \mathbf{SE}$ or vice versa.

Informally, the meaning of the axiom is that ordering between preference differences ("intervals") is preserved irrespective of the "measuring rods" used to measure them. However, contrary to the additive case this does not hold on all X, but only when either points involved in all four relations lie in a single 3C-cone, or points involved in two relations lie in one 3C-cone and those involved in the other two in another.

A5. Whenever $ap \preccurlyeq bq, cp \succcurlyeq dq$ and $ay_0 \sim x_0\pi(a), by_0 \sim x_0\pi(b), cy_1 \sim x_1\pi(c), dy_1 \sim x_1\pi(d)$, and also $e\pi(a) \succcurlyeq f\pi(b)$, it holds $e\pi(c) \succcurlyeq f\pi(d)$, for all $ap, bq, cp, dq \in \mathbf{NW}$ or $\mathbf{SE}$ provided X_1 is essential on the subset which contains these points, $ay_0, by_0, cy_1, dy_1 \in \mathbf{NW}$ or $\mathbf{SE}$, $x_0\pi(a), x_0\pi(b), x_1\pi(c), x_1\pi(d) \in \mathbf{NW}$ or $\mathbf{SE}$ provided X_2 is essential on the subset which contains these points, $e\pi(a), f\pi(b), e\pi(c), f\pi(d) \in \mathbf{NW}$ or $\mathbf{SE}$. Same condition holds for the other dimension symmetrically.

The formal statement of the **A5** is rather complicated, but it simply means that the ordering of the "intervals" is preserved across dimensions. Together with **A4** the conditions are similar to Wakker's trade-off consistency condition [Wakker, 1991b] . The axiom bears even stronger similarity to Axiom 5 (compatibility) from section 8.2.6 of [Krantz et al., 1971]. Roughly speaking, it says that if the "interval" between c and d is "larger" than that between a and b, then "projecting" these intervals onto another dimension by means of equivalence relations must leave this order unchanged. We additionally require the comparison of intervals and "projection" operations to be consistent - meaning that quadruples of points in each part of the statement lie in the same 3C-cone. Another version of this axiom can be formulated in terms of standard sequences (similarly to axiom 5' in Krantz et al. [1971]).

A6. Bi-independence : Let $ap, bp, cp, dp \in \mathbf{SE}(\mathbf{NW})$ and $ap \succ bp$. If for some $q \in X_2$ also exist $cq \succ dq$, then $cp \succ dp$. Symmetric condition holds for the second coordinate.

This is a necessary condition as shown in the following example. Assume $ap, bp, cp, dp \in \mathbf{SE}$ and $ap \succ bp, cp \sim dp$. Assume also there exist $cq, dq \in \mathbf{NW}$ such that $cq \succ dq$. Then, provided the representation exists, we get

$$\alpha_1 f_1(a) + \alpha_2 f_2(p) > \alpha_1 f_1(b) + \alpha_2 f_2(p)$$
$$\alpha_1 f_1(c) + \alpha_2 f_2(p) = \alpha_1 f_1(d) + \alpha_2 f_2(p)$$
$$\beta_1 f_1(c) + \beta_2 f_2(q) > \beta_1 f_1(d) + \beta_2 f_2(q).$$

The first inequality entails $\alpha_1 \neq 0$. From this and the following equality follows $f_1(c) = f_1(d)$, which contradicts with the last inequality. Thus $cq \succ dq$ implies $cp \succ dp$ but only in the presence of $ap \succ bp$ in the same "region" ($\mathbf{SE}$ or $\mathbf{NW}$). This is also the reason behind the name we gave to this condition - "bi-independence". Together with the structural assumption (below), bi-independence also implies some sort of "comonotonic strong monotonicity" Wakker [1989].

Lemma 1. *If coordinate 1 is essential on* $\mathbf{SE}(\mathbf{NW})$, $a \succcurlyeq_1 b$ *iff* $ap \succ bp$ *for all* $ap, bp \in \mathbf{NW}$. *Symmetrical statement holds for coordinate 2.*

Conceptually, Lemma 1 implies that if a coordinate is essential on some 3C-cone $\mathbf{NW}^z(\mathbf{SE}^z)$, then it is also essential on $\mathbf{NW}^x(\mathbf{SE}^x)$ for all $x \in \mathbf{NW}(\mathbf{SE})$. This allows us to make statements like "coordinate i is essential on $\mathbf{NW}$".

A7. Both coordinates are essential on X.

A8. Restricted solvability : if $x_i a_j \succcurlyeq y \succcurlyeq x_i c_j$, then there exists $b_j : x_i b_j \sim y$, for $i, j \in \{1, 2\}$.

A9. Archimedean axiom: for every $z \in \mathbf{NW}(\mathbf{SE})$ every bounded standard sequence contained in $\mathbf{NW}^z(\mathbf{SE}^z)$ is finite.

Structural assumption. For no $a, b \in X_1$ holds $ap \sim bp$ for all $p \in X_2$. Similarly, for no $p, q \in X_2$ it holds $ap \sim aq$ for all $a \in X_1$. If such points exist, say $ap \sim bp$ for all $p \in X_2$, then we can build the representation for a set $X_1' \times X_2$ where $X_1' = X_1 \setminus a$, and later extend it to X by setting $f_1(a) = f_1(b)$.

Dense-rangedness. We assume that whenever $a_i p_j \succ b_i p_j$ there exits $c_i \in X_i$ such that $a_i p_j \succ c_i p_j \succ b_i p_j$, for $i, j \in N$ (X is order dense).

4 Representation theorem

Theorem 1. *Let $\succcurlyeq$ be an order on X and let X be order dense and the structural assumption hold. Then, if axioms **A1-A9** are satisfied, there exist a uniquely determined capacity ν and value functions $f_1 : X_1 \to \mathbb{R}, f_2 : X_2 \to \mathbb{R}$, such that $\succcurlyeq$ can be represented by the Choquet integral:*

$$x \succcurlyeq y \iff C(\nu, (f_1(x_1), f_2(x_2))) \geq C(\nu, (f_1(y_1), f_2(y_2))), \tag{1}$$

for all $x, y \in X$. Value functions have the following uniqueness properties:

1. *If $\nu(\{1\}) + \nu(\{2\}) = 1$, then for any functions $g_1 : X_1 \to \mathbb{R}, g_2 : X_2 \to \mathbb{R}$ such that (1) holds with f_i substituted by g_i, it holds $f_i(x_i) = \alpha g_i(x_i) + \beta_i$.*
2. *If $\nu(\{1\}) \in (0,1)$ and $\nu(\{2\}) \in (0,1)$ and $\nu(\{1\}) + \nu(\{2\}) \neq 1$, then for any functions $g_1 : X_1 \to \mathbb{R}, g_2 : X_2 \to \mathbb{R}$ such that (1) holds with f_i substituted by g_i, it holds $f_i(x_i) = \alpha g_i(x_i) + \beta$.*
3. *If $\nu(\{2\}) \in (0,1)$, $\nu(\{1\}) \in \{0,1\}$, then for any functions $g_1 : X_1 \to \mathbb{R}, g_2 : X_2 \to \mathbb{R}$ such that (1) holds with f_i substituted by g_i, it holds :*
   - *$f_i(x_i) = \alpha g_i(x_i) + \beta$, for all x such that $f_1(x_1) < \max f_2(x_2)$ and $f_2(x_2) > \min f_1(x_1)$;*
   - *$f_i(x_i) = \psi_i(g_i(x_i))$ where ψ_i is an increasing function, otherwise.*
4. *If $\nu(\{2\}) \in \{0,1\}$, $\nu(\{1\}) \in (0,1)$, then for any functions $g_1 : X_1 \to \mathbb{R}, g_2 : X_2 \to \mathbb{R}$ such that (1) holds with f_i substituted by g_i, it holds :*
   - *$f_i(x_i) = \alpha g_i(x_i) + \beta$, for all x such that $f_2(x_2) < \max f_1(x_1)$ and $f_1(x_1) > \min f_2(x_2)$;*
   - *$f_i(x_i) = \psi_i(g_i(x_i))$ where ψ_i is an increasing function, otherwise.*
5. *If $\nu(\{1\}) = \nu(\{2\}) = 0$ or $\nu(\{1\}) = \nu(\{2\}) = 1$, then for any functions $g_1 : X_1 \to \mathbb{R}, g_2 : X_2 \to \mathbb{R}$ such that (1) holds with f_i substituted by g_i, it holds : $f_i(x_i) = \psi_i(g_i(x_i))$ where ψ_i are increasing functions such that $f_1(x_1) = f_2(x_2) \iff g_1(x_1) = g_2(x_2)$.*

5 Proof sketch

Many aspects of the proof are similar to the characterization in [Wakker, 1991a]. The critical difference is in step 10, where it is shown that value functions for different coordinates are equal for the points from the set Θ.

1. Show that $\mathbf{SE}(\mathbf{NW})$ can be covered by sets $\mathbf{SE}^z(\mathbf{NW}^z)$ with $z \in \Theta$.

2. Excluding "extreme" elements of Θ, i.e. points which have maximal or minimal coordinates, show that for any $z \in \Theta$ there exists an additive representation for $\succcurlyeq$ on $\mathbf{SE}^z$ and $\mathbf{NW}^z$.
3. Having built additive representations for $\succcurlyeq$ on $\mathbf{SE}^{z_1}$ and $\mathbf{SE}^{z_2}$, show that there exists an additive representation on $\mathbf{SE}^{z_1} \cup \mathbf{SE}^{z_2}$.
4. Show that this representation - call it V^{SE}, can be extended to cover all $\mathbf{SE}$ (by "joining" representations for all $z \in \Theta$).
5. Perform steps 2 and 3 for **NW** and obtain V^{NW}.
6. Align and scale V^{SE} and V^{NW} such that $V_1^{SE} = V_1^{NW}$ on the common domain, and $V_2^{SE} = \lambda V_2^{NW}$ on their common domain.
7. Pick two points r^0, r^1 from Θ and set r^0 as a common zero. Set $V_1^{SE}(r_1^1) = 1$ and define $\phi_1(x_1) = V^{SE}(x_1), \phi_2(x_2) = V^{SE}(x_2)/V^{SE}(r_2^1)$.
8. Representations now are $\phi_1 + k\phi_2$ on $\mathbf{SE}$ and $\phi_1 + \lambda k \phi_2$ on $\mathbf{NW}$.
9. Rescale so that factors sum up to one : $\frac{1}{1+k}\phi_1 + \frac{k}{1+k}\phi_2$, $\frac{1}{1+\lambda k}\phi_1 + \frac{\lambda k}{1+\lambda k}\phi_2$
10. Show that for all $x \in X$ it holds $\phi_1(x_1) = \phi_2(x_2)$ iff $x \in \Theta$.
11. Extend the representation to the "extreme" elements of Θ.
12. Show that $\succcurlyeq$ can be represented on X by these two representations.
13. Show that $\succcurlyeq$ can be represented by the Choquet integral.

6 Implied commensurateness

We do not assume any commensurateness between elements of criteria sets. Nevertheless, it seems that such commensurateness is implied by the axioms, unless $\succcurlyeq$ can be represented by an additive function. The uniqueness part of Theorem 1 states that for the case of two essential variables the value functions not only have the same unit (as in the additive case), but also the same origin ($f_i(x_i) = \alpha g_i(x_i) + \beta$). In case when $\succcurlyeq$ on $\mathbf{SE}$ and $\mathbf{NW}$ has only one essential variable (different variable on each subset), the following property holds: $f_1(x_1) = f_2(x_2) \iff g_1(x_1) = g_2(x_2)$. Formally, we are mostly interested in points where $\phi_1(x_1) = \phi_2(x_2)$, which turn out to correspond (apart from some extreme cases) to the elements of the set Θ.

Lemma 2. *If $ap \in \Theta$ then for no $b \in X_1$ holds $bp \in \Theta$ and also for no $q \in X_2$ holds $aq \in \Theta$, unless $\succcurlyeq$ has an additive representation.*

Theorem 2. *The following statements hold:*

- *If both $\mathbf{NW}$ and $\mathbf{SE}$ have two essential variables, then for all $x \in X$:*

  $$x \in \Theta \iff \phi_1(x_1) = \phi_2(x_2),$$

  unless $\succcurlyeq$ can be represented by an additive function.
- *If only $\mathbf{NW}$ or only $\mathbf{SE}$ have two essential variables, then for all non-extreme $x \in X$:*

  $$x \in \Theta \Rightarrow \phi_1(x_1) = \phi_2(x_2),$$

  and for all $x \in X$:

  $$\phi_1(x_1) = \phi_2(x_2) \Rightarrow x \in \Theta,$$
- *If both $\mathbf{NW}$ and $\mathbf{SE}$ have only one essential variable, then for all $x \in X$:*

  $$x \in \Theta \iff \phi_1(x_1) = \phi_2(x_2).$$

7 Conclusion

We have proposed a conjoint axiomatization of the Choquet integral for a heterogeneous product set with two dimensions. No commensurateness between dimensions was assumed, rather it was implied by other axioms. We find the interpretation of the implied commensurateness in MCDA terms to be a rather difficult and interesting task, which can probably lead to new results related to the notion of criteria importance. The extension of our results to the n-dimensional case also seems to be an interesting technical problem, which we intend to resolve in subsequent publications.

References

D. Bouyssou, T. Marchant, and M. Pirlot. A conjoint measurement approach to the discrete Sugeno integral. *The Mathematics of Preference, Choice and Order*, pages 85–109, 2009.

D. Bouyssou, M. Couceiro, C. Labreuche, J.-L. Marichal, and B. Mayag. Using choquet integral in machine learning: What can mcda bring? In V. Mousseau and M. Pirlot, editors, *Proceedings DA2PL, "From Multiple Criteria Decision Aid to Preference Learning"*, pages 41–47, November 2012.

M. Grabisch and C. 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):1–44, 2008. ISSN 1619-4500.

S. Greco, B. Matarazzo, and R. Słowiński. Axiomatic characterization of a general utility function and its particular cases in terms of conjoint measurement and rough-set decision rules. *European Journal of Operational Research*, 158(2):271–292, 2004. ISSN 0377-2217.

V. Köbberling and P. P. Wakker. Preference foundations for nonexpected utility: A generalized and simplified technique. *Mathematics of Operations Research*, 28(3):395–423, 2003.

D. H. Krantz, R. D. Luce, P. Suppers, and A. Tversky. Foundation of measurement. vol. 1: Additive and polynomial representations, 1971.

C. Labreuche. An axiomatization of the choquet integral and its utility functions without any commensurability assumption. In S. Greco, B. Bouchon-Meunier, G. Coletti, M. Fedrizzi, B. Matarazzo, and R. R. Yager, editors, *Advances in Computational Intelligence*. 2012.

P. Wakker. Additive representations of preferences, a new foundation of decision analysis; the algebraic approach. In *Mathematical Psychology*, pages 71–87. Springer, 1991a.

P. Wakker. Additive representations on rank-ordered sets. i. the algebraic approach. *Journal of Mathematical Psychology*, 35(4):501–531, 1991b.

P. P. Wakker. *Additive representations of preferences: A new foundation of decision analysis*. Kluwer Academic Publishers Dordrecht, 1989.

Choquistic Utilitaristic Regression

Ali Fallah Tehrani[1], **Christophe Labreuche**[2], **Eyke Hüllermeier**[3]

Abstract. Preference models often represent a (global) degree of utility of an alternative in terms of an aggregation of several local utility degrees, each of which pertains to a specific criterion. Methods for preference learning, i.e., for learning preference models from observed preference data, have mainly focused on fitting the aggregation function while assuming the local utility functions to be given. Taking inspiration from multi-criteria decision aid, this paper makes a first step toward learning both parts of the model simultaneously, the local utility functions and their aggregation into a global degree of utility. More specifically, we consider this problem for two aggregation functions and related machine learning methods, namely linear functons (logistic regression) and the Choquet integral (choquistic regression). Moreover, we also present premilinary experimental results.

1 Introduction

Preference Learning (PL) is an emerging subfield of machine learning which aims at learning preference models from observed preference data [7]. Like machine learning in general, PL is typically focusing on learning highly accurate predictive models from large and possibly noisy empirical datasets. The models used for this purpose are often of generic nature (for example, linear or kernel functions).

As opposed to this, Decision Aid (DA) in general and Multi-Criteria Decision Aid (MCDA) in particular mainly focus on the definition and the analysis of decision models. The analysis of models is obtained by experimental studies or axiomatic approaches [18, 25, 22, 14]. There is a wide variety of decision models studied by DA. In the spirit of Multi-Attribute Utility Theory (MAUT), one can mention very simple models, such as the additive utility model or the weighted sum, but also more elaborate models such as the Choquet integral. The main interest of the latter is its ability to capture interactions among criteria. Another well-developed research area in DA concerns the elicitation of the decision maker's preferences, and has led to the design of elaborate elicitation methods.

In this paper, we consider four approaches in MCDA: **WS**, the weighted sum model, where the weights assigned to criteria are to be determined; **AU**, the additive utility model, where a (local) utility function has to be determined for every attribute/criterion; **Ch**, the Choquet integral, where the capacity over the set of criteria needs to be determined; $\mathbf{Ch}+\mathbf{U}$, the combination of a Choquet integral and utility functions, where the capacity and the utility functions are determined simultaneously.

There is evidence in MCDA showing the importance of learning not only weights on criteria but also a proper scaling and normalization of these criteria, that is, a local utility function for each individual criterion [4, 19]. This provides a strong motivation for the approaches **AU** and $\mathbf{Ch}+\mathbf{U}$.

We are interested in the counterparts of the above four models in a machine learning context. More specifically, we consider the simple case in which the overall evaluation of an alternative can only assume two values, i.e., each alternative is categorized as “good” or “bad”. From a machine learning point of view, the problem can thus be tackled by means of binary classification techniques. Extension to problems such as ordinal classification or different types of ranking problems are left for future work.

The problems of identifying the **WS** and **Ch** models have been tackled in PL based on the maximum likelihood principle for model estimation. While the former leads to conventional logistic regression as a learning method, a generalization called choquistic regression [24] has been proposed for the latter.

Surprisingly, there is no counterpart of the approaches **AU** and $\mathbf{Ch}+\mathbf{U}$ in PL so far. We propose such an extension in this paper. In PL, the features/criteria describing an alternative are usually normalized before learning the model, using simple techniques such as standardization. The idea of this paper is to learn not only weights of features or feature subsets, but also a suitable scaling of the features.

The paper is organized as follows. The next two sections provide some background on the Choquet integral and the classification methods of logistic and choquistic regression. In Section 4, the **WS**, **AU**, **Ch** and $\mathbf{Ch}+\mathbf{U}$ models are discussed. In Section 5, we introduce our approach to the combined learning of the Choquet integral and local utility functions. Finally, Section 6 presents preliminary experimental results.

2 Background on the Choquet Integral

Let $M = \{1, \ldots, m\}$ be the set of elements (that shall correspond to *criteria* in DA and to attributes/features in a machine learning context later on). A *capacity* [5] (also called *fuzzy measure* [23]) on M is a set function $\mu : 2^M \to \mathbb{R}$ such that

- normalization: $\mu(\emptyset) = 0$, $\mu(M) = 1$,
- monotonicity: $\mu(A) \leq \mu(B)$ whenever $A \subseteq B \subseteq M$.

[1] Department of Mathematics and Computer Science, University of Marburg, Germany

[2] Thales Research & Technology, 91767 Palaiseau cedex, France

[3] Department of Computer Science, University of Paderborn, Germany

Roughly speaking, $\mu(A)$ is the importance of criteria in A.

We introduce an important linear transformation over capacities. The *Möbius transform* [20] of a capacity μ, denoted by $\mathbf{m}$, is the unique solution of the equation

$$\mu(A) = \sum_{B \subseteq A} \mathbf{m}(B), \quad \forall A \subseteq M,$$

given by

$$\mathbf{m}(A) = \sum_{B \subseteq A} (-1)^{|A \setminus B|} \mu(B).$$

A capacity μ is *k-additive* [9] if its Möbius transform satisfies $\mathbf{m}(A) = 0$ for all $A \subseteq M$ such that $|A| > k$, and there exists $A \subseteq M$, $|A| = k$, such that $\mathbf{m}(A) \neq 0$. We are particularly interested on 2-additive capacity as it represents interaction between pairs of criteria. The main interest of k-additive capacities is that the number of unknowns is $\sum_{i=0}^{k} \left(\begin{array}{c} m \\ i \end{array} \right)$, which is much less that the 2^m parameters of a capacity μ, when k is small.

The normalization and monotonicity conditions on μ can be turned into monotonicity conditions on $\mathbf{m}$

$$\mathbf{m}(\emptyset) = 0 \quad , \quad \sum_{A \subseteq M} \mathbf{m}(A) = 1 \tag{1}$$

$$\sum_{B \subseteq A} \mathbf{m}(B \cup \{i\}) \geq 0 \quad \forall A \subset M, \forall i \in M \setminus A \tag{2}$$

Now, consider a function $f : M \to \mathbb{R}$. The *Choquet integral* prescribes how to integrate/aggregate f w.r.t. a non-additive measure μ [5]. The Choquet integral of f can be written w.r.t. the Möbius coefficients $\mathbf{m}$ as follows:

$$\mathcal{C}_{\mathbf{m}}(f) := \sum_{A \subseteq M} \mathbf{m}(A) \min_{i \in A} f(i) = C_\mu(f) \ .$$

For a 2-additive capacity, we have

$$\mathcal{C}_{\mathbf{m}}(f) = \sum_{i \in M} \mathbf{m}(\{i\}) f(i) + \sum_{\{i,j\} \subseteq M} \mathbf{m}(\{i,j\}) \min(f(i), f(j)) \ .$$

This expression will be called 2-additive Choquet integral in this paper.

3 Background on Classification

This section recalls some background on classification in general and logistic and choquistic regression is particular.

3.1 Classification

Consider a classification problem, that is, the problem of learning a model $\mathcal{L} : \mathcal{X} \to \mathcal{Y}$ that maps instances $\boldsymbol{x} \in \mathcal{X}$ to categories $y \in \mathcal{Y}$, where $\mathcal{Y}$ is assumed to be a finite set of class labels. As mentioned before, we consider the case of binary classification, where $\mathcal{Y} = \{0, 1\}$ is composed of two classes, the negative (0) and the positive (1) one. Instances $\boldsymbol{x}$ are typically characterized in terms of a feature fector, i.e., by a value on each of a predefined set of features:

$$\boldsymbol{x} = (x_1, \ldots, x_m) \in \mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_m.$$

Given a set of training data

$$\mathcal{D} = \left\{ (\boldsymbol{x}^{(j)}, y^{(j)}) \right\}_{j=1,\ldots,n} \subseteq (\mathcal{X} \times \mathcal{Y})^n \ , \tag{3}$$

the aim is to learn a classifier $\mathcal{L}$ with an as low as possible risk

$$R(\mathcal{L}) = \int_{\mathcal{X} \times \mathcal{Y}} L(\mathcal{L}(\boldsymbol{x}), y) \, d\mathbf{P}_{\mathcal{X}\mathcal{Y}}(\boldsymbol{x}, y) \ ,$$

where L is a loss function and $\mathbf{P}_{\mathcal{X}\mathcal{Y}}$ is an (unknown) probability measure on $\mathcal{X} \times \mathcal{Y}$ modeling the data generating process. The simple 0/1 loss function is defined as $L(\widehat{y}, y) = 0$ if $\widehat{y} = y$, and $L(\widehat{y}, y) = 1$ otherwise.

3.2 Logistic Regression (LR)

Logistic regression models the probability of the positive class (and hence of the negative class) as a linear (affine) function of the input attributes. More specifically, since a linear function does not necessarily produce values in the unit interval, the response is defined as a generalized linear model, namely in terms of the logarithm of the probability ratio:

$$\log \left(\frac{\mathbf{P}(y = 1 | \boldsymbol{x})}{\mathbf{P}(y = 0 | \boldsymbol{x})} \right) = \beta + \boldsymbol{\omega}^\top \boldsymbol{x} \ , \tag{4}$$

where $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_m) \in \mathbb{R}^m$ is a vector of regression coefficients and $\beta \in \mathbb{R}$ is a bias term (intercept). As $\mathbf{P}(y = 0 | \boldsymbol{x}) = 1 - \mathbf{P}(y = 1 | \boldsymbol{x})$, we obtain

$$\pi_{LR}(\boldsymbol{x}) = \mathbf{P}(y = 1 | \boldsymbol{x}) = \frac{1}{1 + \exp\big(- \beta - \boldsymbol{\omega}^\top \boldsymbol{x} \big)} \ . \tag{5}$$

Estimation of the parameters $\boldsymbol{\omega}$ and β is done on the basis of the training data (3), typically via maximizing the log-likelihood function

$$\begin{aligned} l(\beta, \boldsymbol{\omega}) &= \sum_{j=1}^{n} \log \mathbf{P}\big(y^{(j)} | \boldsymbol{x}^{(j)}, \beta, \boldsymbol{\omega}\big) \\ &= \sum_{j=1}^{n} y^{(j)} \ \log \pi_{LR}(\boldsymbol{x}^{(j)}) + \sum_{j=1}^{n} (1 - y^{(j)}) \ \log(1 - \pi_{LR}(\boldsymbol{x}^{(j)})) \ . \end{aligned}$$

3.3 Choquistic Regression (CR)

In choquistic regression (CR), the linear term $\boldsymbol{\omega}^\top \boldsymbol{x}$ in (4) is replaced by a Choquet integral [24]. In order to apply the Choquet integral, the values of attributes are normalized with the help of a function $f_{\boldsymbol{x}} : M \to [0, 1]$, indexed by $\boldsymbol{x}$. Then (4) is transformed into

$$\log \left(\frac{\mathbf{P}(y = 1 | \boldsymbol{x})}{\mathbf{P}(y = 0 | \boldsymbol{x})} \right) = \gamma\big(\, C_\mu(f_{\boldsymbol{x}}) + \beta \big) \ , \tag{6}$$

where γ is a multiplicative factor. Here, $\gamma > 0$ and β are real constants. The γ-parameter is called precision parameter and β is the intercept. More details about theses parameters and their interpretation can be found in [24].

4 Existing Models and Elicitation Techniques in DA

In DA, the decision maker (DM) is supposed to have preferences over a given set of alternatives, which correspond to the set of instances $\mathcal{X}$ in a machine learning context. These preferences are expressed in terms of a binary relation $\succsim$, which is reflexive and transitive (possibly complete). The fundamental problem of decision theory is to build a numerical representation of $\succsim$. The most classical representation of $\succsim$ is the decomposable form [14]

$$\boldsymbol{x} \succsim \boldsymbol{x}' \iff U(\boldsymbol{x}) \geq U(\boldsymbol{x}') \qquad \forall \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X},$$

where

$$U(\boldsymbol{x}) = F(u_1(x_1), \ldots, u_m(x_m)) \ , \tag{7}$$

$F : \mathbb{R}^m \to \mathbb{R}$ is an aggregation function [12], $u_i : \mathcal{X}_i \to \mathbb{R}$ $(i = 1, \ldots, m)$ are called *utility functions* or *value functions.* Utility functions are consistent with partial order $\succsim_i$, i.e.,

$$x_i \succsim_i x_i' \iff u_i(x_i) \geq u_i(x_i') \quad \forall x_i, x_i' \in \mathcal{X}_i \ . \tag{8}$$

As mentioned before, methods in PL have focused on learning the aggregation function F so far, i.e., the global utility U, while assuming the (local) utility functions u_i to be given.

We consider two aggregation functions F here: the weighted sum and the Choquet integral. Several elicitation techniques have been developed in the DA community for model (7) with these two aggregation functions, mainly based on linear programming. These techniques (called **WS**, **AU**, **Ch** and **Ch + U**) are described in the remainder of this section.

4.1 Weighted Sum (WS)

The simplest model is the weighted sum $U(\boldsymbol{x}) = \sum_{i \in M} \omega_i \, u_i(x_i)$, where ω_i is the weight of feature i. As for the Choquet integral, the utility function u_i need to return comparable values. Formally, the scales represented by two utility functions u_i and u_j shall be *commensurate.* We say that two scales u_i, u_j over criteria i and j are *commensurate* if for every $x_i \in \mathcal{X}_i, x_j \in \mathcal{X}_j$ such that $u_i(x_i) = u_j(x_j)$, the degrees of satisfaction felt by the DM on criteria i and j are equal. It is often assumed that the utilities lie in the interval $[0, 1]$, where 1 (resp. 0) means that the criterion is completely satisfied (unsatisfied).

In DA, the weights and the utility functions are usually learnt in two steps. Firstly, each utility function u_i is constructed separately from learning data restricted to $\mathcal{X}_i$ (intra-feature learning), using methods such as AHP [21] or MACBETH [3, 2]. Then, assuming the utility functions to be given, the weights $\boldsymbol{\omega}$ are learnt from additional data, which typically consists of pairwise comparisons of alternatives [3, 2]. To this end, techniques such as linear programming (LP) can be used. The determination of weights in a weighted sum with fixed utility functions is called **WS** (Weighted Sum) in this paper.

4.2 Additive Utility (AU)

The main drawback of the previous approach is that the weights and the utility functions are not learnt simultaneously. In particular, the first step of the construction of utility functions is very important, since the weights $\boldsymbol{\omega}$ cannot compensate a bad choice of these functions. It may happen that the training data is not representable by the weighted sum when the utility functions are fixed, but could be represented by a weighted sum if both the weights and the utility functions are identified at the same time. Yet, the problem of finding $\boldsymbol{\omega}$ and the $u_1, \ldots, u_m$ simultaneously is no longer linear and therefore complex to solve.

In order to bypass this difficulty, it is more convenient to rewrite the weighted sum as an additive utility function $U(\boldsymbol{x}) = \sum_{i \in M} v_i(x_i)$ (with $v_i(x_i) = \omega_i \, u_i(x_i)$). Unlike the weighted sum, the value functions v_i are not commensurate. In particular, they are neither normalized nor restricted to the interval $[0, 1]$. The UTA (UTilities Additives) method can learn all utility functions at the same time using an LP approach applied to comparisons of alternatives [13]. The approach of constructing all utilities at the same time is called **AU** (Additive Utility).

4.3 Choquet Integral (Ch)

The model of the weighted sum (or additive utility) exhibits a limited expressive power as it assumes independence among criteria. A more versatile model is the Choquet integral w.r.t. a capacity: $U(\boldsymbol{x}) = C_\mu(u_1(x_1), \ldots, u_n(x_n))$, where μ is a capacity on M. The utility functions need to be commensurate and are often normalized to $[0, 1]$.

As for the weighted sum, the capacity μ and the utility functions u_i are typically learnt in two steps [17, 11]. An extension of the MACBETH approach has been proposed in [17] to construct each utility function u_i separately without the knowledge of μ. Then, given the utility functions, many papers propose algorithms to construct the capacity, transforming training data into an optimization problem (see [10, 11] for review). Most often, the capacity μ is learnt using LP or quadratic programming. The construction of the capacity from fixed utility functions is called **Ch** (Choquet integral).

4.4 Choquet Integral and Utility Functions (Ch + U)

Like for the weighted sum, one may wonder whether it is relevant to learn the utility functions and the capacity at the same time. There are two references emphasizing the importance of doing so. The first one explains that when the utility functions u_i are fixed, it is very easy to construct examples of preferences that cannot be represented by the Choquet integral, whereas if both the u_i and the capacity μ can be tuned at the same time, the construction of such examples becomes much more complex [4]. The second reference presents an experimental study comparing the representativeness of models **WS**, **AU** and **Ch** [19]. These authors measure the number of datasets that can be represented by **WS**, **AU** and **Ch**, where the datasets are random orders on randomly generated instances. In this experiment, the **AU** model tends to represent the random datasets better than **Ch** [19]. From this experiment, we conclude that the Choquet integral might not be very useful unless suitable utility functions u_i are provided.

Very few theoretical works can be found on MCDA models composed of both the Choquet integral and its utility functions [15, 16]. The determination of not only the admissible capacities but also the utility functions has been considered from a practical side only in two papers [1, 8]. A stochastic method (Monte Carlo or genetic algorithm) has been proposed in [1], and a fixed-point approach using nested linear programs is used in [8].

We have justified the importance of learning the capacity and the utility functions at the same time. This approach is called $\mathbf{Ch}+\mathbf{U}$ (Choquet integral and utility functions).

The previous discussion has shown that, in DA, methods like $\mathbf{Ch}+\mathbf{U}$ are typically motivated by their representative power. Here, the underlying assumption is that the more preference relations a model can represent, the "better" it is. Therefore, it is very important to mention that, in a machine learning context, a high level of expressivity of a model class is not a desirable property per se. On the contrary, a higher level of expressivity will in general increase the danger of overfitting the training data, and hence of generalizing poorly beyond that data. In fact, the main problem in machine learning is to find a model class with the right level of expressivity, neither too low nor too high. Nervertheless, it is of course interesting to assess the performance of $\mathbf{Ch}+\mathbf{U}$ compared to **Ch**, **AU** and **WS** in a machine learning context.

Figure 1 summarizes the relationship between the four models discussed so far (arrows depict inheritance).

- **Ch** encompasses **WS**, since a weighted sum is a particular case of a Choquet integral. As utilities are fixed both in **WS** and **Ch**, the latter is strictly more general than the former.
- **AU** is more general than **WS**, since the additive utility model (where utilities are to be determined) encompasses the weighted sum (where utilities are fixed).
- $\mathbf{Ch}+\mathbf{U}$ amounts to identify a capacity and the utility functions simultaneously, and is thus more general than **AU**, which does not allow for any interaction among criteria.
- Likewise $\mathbf{Ch}+\mathbf{U}$ is more general than **Ch**, since the utilities are fixed in this latter.

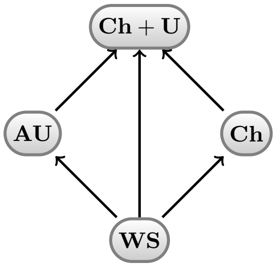


Figure 1. Relationships between the models.

5 Utilitaristic (Choquistic) Regression

5.1 Models used for Classification

We consider classification problems in which each attribute domain $\mathcal{X}_i$ is equipped with a natural order $\succsim_i$, either "*the larger the better*" or "*the smaller the better*". For the sake of clarity, we shall subsequently distinguish between the (original) *feature or attribute value* x_i of an instance $\boldsymbol{x}$ and the corresponding *utility degree* $u_i(x_i)$; for example, the value of the feature `price` of an alternative could be $x_i = 99$ €, and the corresponding utility degree $u_i(x_i) = 0.7$. Replaing feature values by utility degrees comes down to defining (or learning) a mapping $\mathcal{X} \to \mathbb{R}$ that we shall refer to as a *scaling* of the attribute/feature; in agreement with our monotonicity assumption, this mapping should be monotonic, i.e., either non-decreasing or non-increasing.

Generalizing (4) and (6) , we are interested in models of the form

$$\log\left(\frac{\pi_U(\boldsymbol{x})}{1-\pi_U(\boldsymbol{x})}\right) = U(\boldsymbol{x}) + \beta \ , \tag{9}$$

where the utility U will depend on the choice of the DA model, and $\pi_U(\boldsymbol{x}) = \mathbf{P}(y=1|\boldsymbol{x})$ is indexed by U. This gives

$$\pi_U(\boldsymbol{x}) = \frac{1}{1+\exp(-\beta - U(\boldsymbol{x}))} \ . \tag{10}$$

The model U contains unknown parameters $\mathbf{p}$ that will be explained below.

The four elicitation models described in Section 4 have natural counterparts in PL (see Table 1 for the correspondence):

- Counterpart of **WS**: The idea of WS is to learn a weight vector on the features. This comes down to standard logistic regression (see Section 3.2), with each input attribute being scaled beforehand. One may think of standardization, i.e.,

  $$u_i = \frac{x_i - m_i}{\sigma_i} \quad \text{or} \quad u_i = \frac{m_i - x_i}{\sigma_i},$$

  depending on whether $\succsim_i$ represents "*the larger the better*" or "*the smaller the better*", where m_i and σ_i are the mean and the standard deviation of the i-th attribute in the training data. Then, the model U is given by

  $$U(\boldsymbol{x}) = \boldsymbol{\omega}^\top \boldsymbol{u}(\boldsymbol{x}) = \sum_{i=1}^m \omega_i \, u_i, \tag{11}$$

  where the unknowns are the weights $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_m)$. The representation of natural preferences $\succsim_i$ yields the following monotonicity conditions on $\boldsymbol{\omega}$:

  $$\omega_1 \geq 0, \ldots, \omega_m \geq 0. \tag{12}$$

  This model will be named **LR** (Logistic Regression).
- Counterpart of **AU**: The idea here is to replace the dot product $\boldsymbol{\omega}^\top \boldsymbol{u}(\boldsymbol{x})$ by a sum of utilities over the features:

  $$U(\boldsymbol{x}) = \sum_{i=1}^m u_i(x_i), \tag{13}$$

  where the utility functions are parametric and the corresponding parameters will be described in Section 5.2. This model will be named **UR** (Utilitaristic Regression).
- Counterpart of **Ch**: The idea of **Ch** is to learn the capacity of a Choquet integral, where the utility functions are already fixed:

  $$U(\boldsymbol{x}) = \gamma \, \mathcal{C}_{\mathbf{m}}(\boldsymbol{u}(\boldsymbol{x})), \tag{14}$$

  where $\boldsymbol{u}(\boldsymbol{x})$ is the same scaling as in **LR**, $\gamma > 0$ and the Möbius coefficients $\mathbf{m}$ satisfy (1) and (2). This is exactly choquistic regression described in Section 3.3. This model will thus be named **CR** (Choquistic Regression).

- Counterpart of **Ch + U**: In the last model, both the capacity and the utility functions are learnt:

$$U(\boldsymbol{x}) = \gamma\, \mathcal{C}_{\mathbf{m}}(u_1(x_1), \ldots, u_m(x_m)), \tag{15}$$

where the utility functions are parametric and will be described in Section 5.2, $\gamma > 0$ and the Möbius coefficients $\mathbf{m}$ satisfy (1) and (2). This model will be named **CUR** (Choquistic Utilitaristic Regression).

Note that the use of a Choquet integral requires the utility functions to be commensurate, an assumption that is not required for the additive utility (13). This is why the utilites u_i in (15) are normalized ($u_i \in [0, 1]$), which is not necessarily the case in (13).

One can easily turn Figure 1 into a figure depicting the relationships between the four methods **LR**, **UR**, **CR** and **CUR** (see Figure 2).

MCDA models	associated methods in PL
WS	**LR**
Ch	**CR**
AU	**UR**
Ch + U	**CUR**

Table 1. Correspondence between MCDA models and the associated methods in PL.

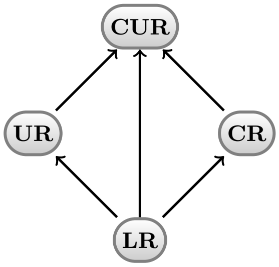


Figure 2. Relationship between the PL methods.

5.2 Representation of the Utility Functions

This section describes how the utility functions are represented in the models **UR** and **CUR**.

In MCDA, utility functions are most of the time considered as piecewise affine [13, 2]. In order to learn utility function u_i, the attribute domain $\mathcal{X}_i$ is discretized. The DM is assumed to provide a finite set $a_i^1, a_i^2, \ldots, a_i^{p_i}$ of p_i distinct elements of $\mathcal{X}_i$, with $a_i^1 < a_i^2 < \cdots < a_i^{p_i}$. The two extreme elements a_i^1 and $a_i^{p_i}$ are reference elements for which criterion i is either completely satisfied (largest value of utility, e.g. 1 for **CUR**) or not satisfied at all (utility 0). If the utility function is non-decreasing ($\succsim_i$ corresponds to $\geq$), then element a_i^1 is the largest element in $\mathcal{X}_i$ which is considered as not satisfactory at all, and $a_i^{p_i}$ is the smallest element in $\mathcal{X}_i$ which is considered as completely satisfactory (see Figure 3). Hence we fix $u_i(a_i^1) = 0$. Value $u_i(a_i^{p_i})$ is fixed to 1 only if the commensurability assumption is made—that is, for the model **CUR**.

In the previous representation of utility function u_i, methods in MCDA assume that $a_i^1, \ldots, a_i^{p_i}$ are given and the unknowns are $u_i(a_i^1), \ldots, u_i(a_i^{p_i})$. The utility function is then obtained for all values in $\mathcal{X}_i$ by interpolation (see Figure 3). The main problem with this representation is that the values of the thresholds $a_i^1, a_i^{p_i}$ is very crucial, as it depicts where improvement in the value of a feature has an impact in the overall utility U. In a machine learning setting, one cannot expect that these thresholds are given by an expert. The next idea would be to assume that the unknowns of the utility functions are $a_i^1, a_i^{p_i}$ and $u_i(a_i^1), \ldots, u_i(a_i^{p_i})$, where $a_i^2, a_i^3, \ldots, a_i^{p_i-1}$ depend linearly on $a_i^1, a_i^{p_i}$. The problem with this representation is that the log-likelihood $l(\beta, \mathbf{p})$ is not continuously differentiable in $a_i^1, a_i^{p_i}$. Hence, it would be very hard for optimization algorithms to maximize $l(\beta, \mathbf{p})$.

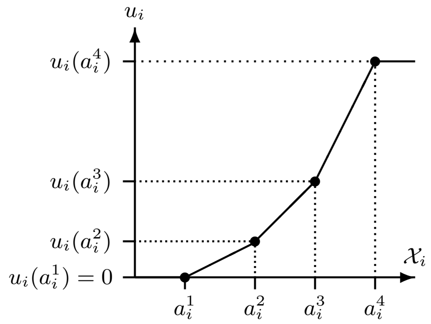


Figure 3. Example of piecewise affine utility function u_i.

We would like to define a parametric expression of u_i, where u_i is bounded, $\mathcal{X}_i$ can be any subset of $\mathbb{R}$ (allowing for both negative and positive values), such that $l(\beta, \mathbf{p})$ is continuously differentiable in the parameters of u_i. A natural choice of a parametric expression is the sigmoid function

$$\frac{1}{1 + \exp(-\eta_i\,(x_i - \lambda_i))}\ , \tag{16}$$

where λ_i is a shift parameter, and η_i is the slope of the sigmoid functions. The sigmoid function is skew-symmetric around the point λ_i. It is easy to see that the elements $x_i \in \mathcal{X}_i$ for which the derivative of the sigmoid function is above a threshold $\varepsilon > 0$ is the interval

$$\left[\lambda_i - \frac{\Delta}{\eta_i}, \lambda_i + \frac{\Delta}{\eta_i}\right]$$

where $\Delta = \log\left(\frac{t}{2} - 1 + \sqrt{\frac{t^2}{4} - t}\right)$ and $t = \frac{\eta_i}{\varepsilon}$. For ε chosen, the previous interval provides the relevant part of $\mathcal{X}_i$, where a modification on the feature value has a significant impact on the utility. The counterpart of this interval for a piecewise affine utility function is $[a_i^1, a_i^{p_i}]$. Hence expression (16) parameterized by λ_i and η_i allows us to learn the part of $\mathcal{X}_i$ which is the most relevant for the classification problem, that is, where the gradient of u_i takes its largest values.

Expression (16) yields values in $[0, 1]$. When commensurability is not required (for model **UR**—see (13)), we cannot enforce the utilities to lie in the unit interval. Hence we consider a new parameter r_i which controls the range of the utility function

$$\frac{r_i}{1 + \exp(-\eta_i\,(x_i - \lambda_i))}\ , \tag{17}$$

where $r_i > 0$.

We already noted that (17) is skew-symmetric around λ_i. Since other shapes of utility functions are sought in practice, we propose a linear combination of sigmoid functions to capture more complex utility functions:

$$u_i(x_i) = \sum_{l=1}^{p_i} \frac{r_i^l}{1+\exp(-\eta_i^l\,(x_i-\lambda_i^l))} \ , \tag{18}$$

where p_i is the number of sigmoid functions used for utility function u_i, and r_i^l controls the relative strengths among the different sigmoid functions. If the utility function u_i is nondecreasing in x_i, then

$$r_i^1 \geq 0, \ldots, r_i^{p_i} \geq 0 \ . \tag{19}$$

For the model **CUR**, we need to have normalized utility functions, which yields the following constraint:

$$\sum_{l=1}^{p_i} r_i^l = 1 \ . \tag{20}$$

We also enforce the constraints on λ_i^l and η_i^l:

$$\eta_i^l > 0 \quad \forall l \in \{1,\ldots,p_i\} \ , \tag{21}$$

$$\min_{j=1,\ldots,n} x_i^{(j)} \leq \lambda_i^1 < \lambda_i^2 < \cdots < \lambda_i^{p_i} \leq \max_{j=1,\ldots,n} x_i^{(j)}. \tag{22}$$

In (22), the values λ_i^l are necessarily located in the observed range of attribute values $[\min_{j=1,\ldots,n} x_i^{(j)}, \max_{j=1,\ldots,n} x_i^{(j)}]$.

5.3 Parameter Learning

For the **CUR** model, the utility functions and the aggregation functions need to be learnt simultaneously. To this end, we make use of a generalization of the approach proposed in [24] based on likelihood maximization.

Our general model (10) contains a set of parameters, namely β and the parameters $\mathbf{p}$ of utility model U. The list of parameters for the four models **LR**, **CR**, **UR**, **CUR** is summarized in Table 2.

model U	parameters $\mathbf{p}$ of U	constraints on $\mathbf{p}$
LR	$\boldsymbol{\omega} = (\omega_1,\ldots,\omega_m)$	$\omega_1 \geq 0,\ldots,\omega_m \geq 0$
CR	$\mathbf{m}$, γ	$\gamma > 0$, (1) and (2)
UR	$\{r_i^l, \eta_i^l, \lambda_i^l\}_{i,l}$	(19), (21), (22)
CUR	$\mathbf{m}$, γ, $\{r_i^l, \eta_i^l, \lambda_i^l\}_{i,l}$	$\gamma > 0$, (1), (2), (19), (20), (21), (22)

Table 2. List of parameters and constraints for the four models

From the training dataset $\mathcal{D}$ in (3), all unknowns (β and the parameters $\mathbf{p}$ of U) are determined by maximizing the log-likelihood:

$$l(\beta,\mathbf{p}) = \log \mathbf{P}(\mathcal{D}|\beta,\mathbf{p}) = \log \prod_{j=1}^{n} \mathbf{P}\big(y^{(j)}|\boldsymbol{x}^{(j)},\beta,\mathbf{p}\big) \tag{23}$$

$$= \sum_{j=1}^{n} y^{(j)}\ \log \pi_U(\boldsymbol{x}^{(j)}) + \sum_{j=1}^{n}(1-y^{(j)})\ \log(1-\pi_U(\boldsymbol{x}^{(j)})) \ .$$

We obtain

$$l(\beta,\mathbf{p}) = -\sum_{j=1}^{n} \log\Big(1+\exp(-\beta-U(\boldsymbol{x}^{(j)}))\Big)$$

$$-\sum_{j=1}^{n}(1-y^{(j)})\ \Big(\beta+U(\boldsymbol{x}^{(j)})\Big) \ . \tag{24}$$

Basically, for the four problems **LR**, **CR**, **UR** and **CUR**, the parameters are identified by maximizing $l(\beta,\mathbf{p})$ over the variables β and $\mathbf{p}$ (see Table 2) under the monotonicity constraints on $\mathbf{p}$ described in Table 2. For a detailed discussion of this optimization problem and its tractability, we refer to [24]. Practically, the `fmincon` function implemented in the optimization toolbox of Matlab has been used. This method is based on a sequential quadratic programming approach.

6 Experiments

6.1 Experimental Setting

We conducted several experiments to investigate the efficiency of our approach. To this end, we used monotone data that is mainly taken from the UCI data repository.[4] As discussed, the scaling of attributes is accomplished through a convex combination of sigmoid functions. In the experiments, we took the same number of sigmoid functions for all attributes, namely with $p_i = 2$ or 3.

For models **CR** and **CUR**, a 2-additive capacity has been used. For models **LR** and **CR**, features are normalized by mapping values x_i to their empirical quantiles. More precisely, for each attribute $i \in M$, we denote by π_i a permutation on $\{1,\ldots,n\}$ ordering the value of instances on attribute i:

$$x_i^{(\pi_i(1))} \leq x_i^{(\pi_i(2))} \leq \cdots \leq x_i^{(\pi_i(n))}.$$

Then, u_i is interpolating between the following values: $u_i\Big(x_i^{(\pi_i(k))}\Big) = \frac{k-1}{n-1}$, for every $k \in \{1,\ldots,n\}$.

6.2 Main Experimental Results

In the experiments, we compare the four methods **LR**, **CR**, **UR** and **CUR**. This way, we can analyse the added value of learning local utility functions (that is, *scaling*, as used in models **UR** and **CUR**) and/or allowing for *interactions* between criteria (as in models **CR** and **CUR**). Table 3 summarizes the performance of the four models on 9 datasets. The classification accuracy is measured by the 0/1 loss and estimated as averages over 20 repetitions of a 5 fold cross validation. The 0/1 loss is noted $\eta_{\mathbf{LR}}$, $\eta_{\mathbf{CR}}$, $\eta_{\mathbf{UR}}$ and $\eta_{\mathbf{CUR}}$ for the four methods.

We note that **CUR** returns the best predictions in 5 out of the 9 datasets (including case Auto-MPG where it is almost similar to **UR**). Moreover, the improvement of **CUR** compared to **LR** is quite significant on datasets CEV and Auto MPG. We interpret these results are evidence for the importance of a combed use of scaling and interaction.

Despite being the most expressive model, **CUR** does not always achieve the best results. This is not surprising, however, since, as mentioned before, more expressive models are not necessarily advantageous from a learning point of view. For instance, **CUR** returns the worst predictions for the ERA dataset. It is interesting to note that, for this dataset, scaling and interaction both have a positive effect when used in isolation, whereas their joint use does not seem to have any further advantage. One can measure the interaction (or synergy) between AN and IF by the following indicator:

$$\eta_{\mathbf{LR}} - \eta_{\mathbf{CR}} - \eta_{\mathbf{UR}} + \eta_{\mathbf{CUR}}$$

[4] `http://archive.ics.uci.edu/ml/`

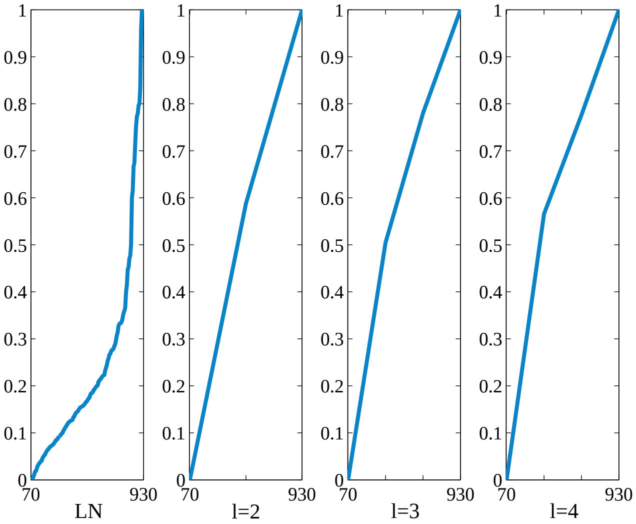


Figure 4. Scaling functions for the criterion `area` on the DenBosch data computed by quantile normalization (called LN here) and local utility learning (where l stands for p_i in the figure).

Note that this indicator is similar to the Shapley interaction index used to interpret a capacity. This indicator is negative when most of the improvement (compared to the baseline **LR**) comes from using either scaling or interaction, but not when using both at the same time (as it is the case for ERA). On the other hand, this indicator is positive when most of the improvement comes when using scaling and interaction simultaneously. The interaction index between scaling and interaction is positive for 5 datasets (CPU, DenBosch, ESL, Mammo, Auto-MPG, Breast Cancer).

Finally, we shall mention that **CUR** induces a complexity in terms of execution time, because of the difficulty to learn at the same time the utility function and the capacity. We will elaborate on this point in future work.

6.3 Case Study

In order to further explore the properties and benefits of the proposed approach to learning utility functions, we carried out an experiment on the DenBosh data. This dataset contains 8 attributes describing houses in the city of Den Bosch: district, area, number of bedrooms, type of house, volume, storeys, type of garden, garage, and price. The output is a binary variable indicating whether the price of the house is low or high (depending on the comparison with a threshold [6]). In the experiments, $p_i = 2$, $p_i = 3$ and $p_i = 4$ are used. We consider the second criteria, namely, area. In Figure 4, the utility functions obtained for this criterion is shown for different methods and parameter configurations. As can be seen, the functions for $p_i = 3$ and $p_i = 4$ are almost the same, suggesting that $p_i = 3$ offers enough flexibility. In Figure 5, the utility functions of all 8 attributes are shown for $p_i = 4$ is chosen.

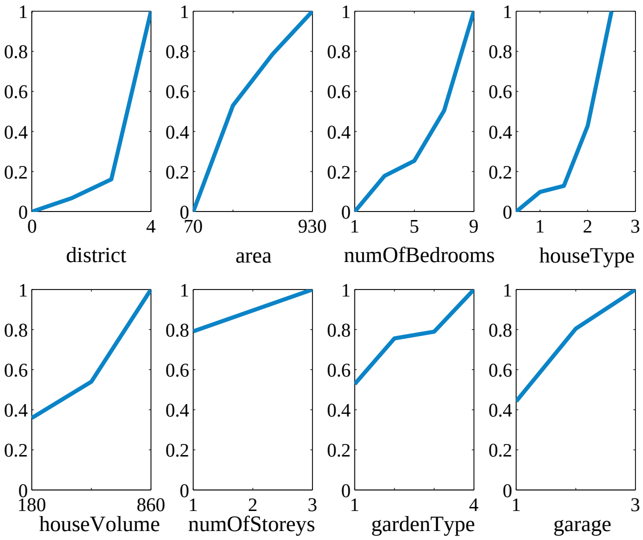


Figure 5. Illustration of attribute scaling on the DenBosch data set (for 2-additive choquistic regression).

7 Conclusion and Future Work

This paper advocates the idea that machine learning techniques can benefit from the use of (monotone) decision models from MCDA. Such models are often expressed in terms of a two-level hierarchy: First, local utility degrees are determined on each criterion, and these utility degrees are then aggregated into an overall evaluation of an alternative. While preference learning methods have focused on learning the aggregation function so far, we highlight the usefulness of simultaneously learning both parts of this hierarchy, not only the aggregation but also the local utility functions.

Two existing methods were extended along this line, namely logistic regression (**LR**) and choquistic regression (**CR**). For the corresponding extensions, called utilitaristic regression (**UR**) and utilitaristic choquistic regression (**UCR**), respectively, utility functions are represented as a linear combination of sigmoid functions. Mathematically, this representation is quite convenient and exhibits several advantages. Our preliminary results on 9 benchmark datasets are promising and suggests the practical interest of local utility learning and representing interaction among features.

In future work, we plan to test other forms of parameterized utility functions, and identify the suitable number of parameters. Moreover, going beyond binary classification, our approach will be extended to other types of preference learning problems.

REFERENCES

[1] S. Angilella, S. Greco, F. Lamantia, and B. Matarazzo. Assessing non-additive utility for multicriteria decision aid. *European Journal of Operational Research*, 158:734–744, 2004.

[2] C. A. Bana e Costa, J.M. De Corte, and J.-C. Vansnick. MACBETH. *International Journal of Information Technology and Decision Making*, 11:359–387, 2012.

[3] C. A. Bana e Costa and J.-C. Vansnick. A theoretical framework for Measuring Attractiveness by a Categorical Based

Datasets	**CUR**, $p_i = 2$	**UCR**, $p_i = 3$	**UR**, $p_i = 2$	**UR**, $p_i = 3$	**LR**	**CR**
ERA	.3191 ± .0185	.3015 ± .0197	**.2894 ±.0239**	.2953 ± .0365	.2932 ± .0261	**.2891 ± .0241**
LEV	.1352 ± .0236	**.1302 ± .0126**	.1415 ±.0190	.1563 ± .0271	.1662 ± .0171	.1500 ± .0207
CEV	.0623 ± .0521	**.0240 ± .0160**	.0583 ±.0153	.0461 ± .0130	.1643 ± .0184	.0719 ± .0091
CPU	**.0285 ± .0301**	**.0244 ± .0252**	.1390 ±.0630	.1171 ± .0549	.0336 ± .0068	.0276 ± .0229
DenBosch	.1630 ± .0859	.1524 ± .0653	.1826 ± .0788	.1884 ± .0807	.1409 ± .0336	**.1283 ± .0683**
ESL	.0660 ± .0196	.0680 ± .0210	.0785 ± .0260	.0670 ± .0312	**.0602 ± .0264**	.0694 ± .0218
Mammo	.1642 ± .0271	**.1553 ± .0317**	.1685 ±.0302	.1600 ± .0303	.1683 ± .0231	.1693 ± .0285
Auto-MPG	**.0038 ± .0084**	.0054 ± .0120	**.0038 ±.0073**	**.0034 ± .0067**	.0538 ± .0282	.0654 ± .0266
Breast Cancer	.2773 ± .0348	.2989 ± .0550	.3079 ±.0635	.3042 ± .0501	**.2669 ± .0483**	.2861 ± .0482

Table 3. The comparison in terms of 0/1 loss between 2-additive Choquistic regression and linear logistic regression equipped by two distinguished normalization methods. The best results for each dataset are highlighted in bold.

Evaluation TecHnique (MACBETH). In *Proc. XIth Int. Conf. on MultiCriteria Decision Making*, pages 15–24, Coimbra, Portugal, August 1994.

[4] D. Bouyssou, M. Couceiro, C. Labreuche, J.-L. Marichal, and B. Mayag. Using choquet integral in machine learning: what can MCDA bring? In *Workshop from Multiple Criteria Decision Aid to Preference Learning*, Mons, Belgium, November 15-16 2012.

[5] G. Choquet. Theory of capacities. *Annales de l'Institut Fourier*, 5:131–295, 1953.

[6] H. Daniels and B. Kamp. Applications of MLP networks to bond rating and house pricing. *Neural Computation and Applications*, 8:226–234, 1999.

[7] J. Fürnkranz and E. Hüllermeier, editors. *Preference Learning.* Springer-Verlag, 2010.

[8] B. Goujon and Ch. Labreuche. Holistic preference learning with the choquet integral. In *Int. Conf. Of the Euro Society for Fuzzy Logic and Technology (EUSFLAT)*, Minano, Italy, 2013.

[9] M. Grabisch. k-order additive discrete fuzzy measures and their representation. *Fuzzy Sets & Systems*, 92:167–189, 1997.

[10] M. Grabisch, I. Kojadinovic, and P. Meyer. A review of capacity identification methods for Choquet integral based multi-attribute utility theory — applications of the Kappalab R package. *Eur. J. of Operational Research*, 186:766–785, 2008.

[11] M. Grabisch and Ch. Labreuche. A decade of application of the Choquet and Sugeno integrals in multi-criteria decision aid. *Annals of Operation Research*, 175:247–286, 2010.

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

[13] E. Jacquet-Lagrèze and Y. Siskos. Assessing a set of additive utility functions for multicriteria decision making: The UTA method. *European J. of Operational Research*, 10:151–164, 1982.

[14] D.H. Krantz, R.D. Luce, P. Suppes, and A. Tversky. *Foundations of measurement*, volume 1: Additive and Polynomial Representations. Academic Press, 1971.

[15] Ch. Labreuche. Construction of a Choquet integral and the value functions without any commensurateness assumption in multi-criteria decision making. In *Int. Conf. Of the Euro Society for Fuzzy Logic and Technology (EUSFLAT)*, Aix Les Bains, France, July 18-22 2011.

[16] Ch. Labreuche. An axiomatization of the Choquet integral and its utility functions without any commensurateness assumption. In *Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU)*, Catania, Italy, July 9-13 2012.

[17] Ch. Labreuche and M. Grabisch. The Choquet integral for the aggregation of interval scales in multicriteria decision making. *Fuzzy Sets & Systems*, 137:11–26, 2003.

[18] Th. Marchant. Towards a theory of MCDM: Stepping away from social choice theory. *Mathematical Social Choice*, 45:343–363, 2003.

[19] M. Pirlot, H. Schmitz, and P. Meyer. 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)*, pages 374–387, Coimbra, Portugal, April 2010.

[20] G. C. Rota. On the foundations of combinatorial theory I. Theory of Möbius functions. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 2:340–368, 1964.

[21] T. L. Saaty. A scaling method for priorities in hierarchical structures. *J. Math. Psychology*, 15:234–281, 1977.

[22] L. J. Savage. *The Foundations of Statistics.* Dover, 2nd edition, 1972.

[23] M. Sugeno. *Theory of fuzzy integrals and its applications.* PhD thesis, Tokyo Institute of Technology, 1974.

[24] A. Fallah Tehrani, W. Cheng, K. Dembczynski, and E. Hüllermeier. Learning monotone nonlinear models using the choquet integral. *Machine Learning*, 89:183–211, 2012.

[25] J. Von Neuman and O. Morgenstern. *Theory of Games and Economic Behavior.* Princeton University Press, 1944.

Hospital rankings: a new challenge for MCDA and preference learning?

Brice Mayag[1]

Abstract. The aim of this paper is to convince the MultiCriteria Decision Aid (MCDA) and Preference Learning communities to investigate and to contribute in the development of methodologies dedicated to hospital ranking. To do so, we present the French hospital ranking and show how these rankings can be built properly through two existing methods: decision tree and ELECTRE Tri.

Key words: MCDA; Machine Learning; Hospital rankings; Decision tree; ELECTRE TRI

1 Introduction

MultiCriteria Decision Aid (MCDA) aims at representing the preferences of a Decision-Maker (DM), or a group of Decision-Makers, over a finite set of alternatives evaluated on several criteria often conflicting. Many softwares implementing MCDA methods have been developed and most of them have proved their efficiency in real applications, e.g. MACBETH [1], MYRIAD [10]. One of the problem statement treated by MCDA is the elaboration of rankings.

Since many years, there exist some hospital rankings published by newspapers. In France, three newspapers publish every year their hospital rankings. In reality they do not evaluate the global hospital, but only its surgery specialties. In our knowledge, two other countries publish regularly hospital rankings:

- *United Stated of America*: these rankings are published each year by a news paper called Usnews[2]. The methodology used is based on the weighted sum and developed by the Research Triangle Institute (RTI international), a scientific organism. The report of 129 pages about this methodology is free available[3].
- *United Kingdom*: the rankings are elaborated by the National Health Service (NHS)[4].

From the view of MCDA, we were interested in the methodologies used in French hospital rankings. We studied them in details, but we were disappointed because all the French methodologies are just presented in few lines (not more than a half page) compared to the Usnews methodology which is presented in more than 100 pages. Furthermore there is no relevant information concerning MCDA aspects. The main reason is that, behind these rankings, there are only journalists (François Malye and Jérôme Vincent for “Le point”) and some very small consulting companies (Le Guide santé for “Le Figaro Magazine” and Santé Value “Le Nouvel Observateur”) without knowledge about good best practices of MCDA. In general, to improve their reputation, the hospitals need and wish to know each year their rank in the published hospital rankings. Most of these hospitals choose to advertise this rank, when they are good, in their website. Health governments agencies also can use these rankings to identify which are the “weak” hospitals.

The challenge we propose here is to use all the scientific background of MCDA to properly structure these real and concrete applications. We propose to identify relevant indicators (criteria) with machine learning methods such as decision tree. The opportunity to test also preference learning algorithms should be investigate. Let us recall that preference learning is a subfield in machine learning in which the goal is to learn a predictive preference model from observed preference information [8]. Because the databases of indicators filled by the French hospitals are public and available under some minor conditions, we can solve this actual problem by giving a valid methodology where algorithms and methods of the two communities are applied.

The paper is organized as follows: we present in Section 2 the three French hospital rankings, especially in weight loss surgery and we give our propositions in Section 3.

2 About French hospital rankings

In France, hospital rankings are published each year by three newspapers: “Le Nouvel observateur”[5], “Le Point”[6] and “Le Figaro Magazine”[7]. To establish these rankings, they manipulate data coming from some official databases like HOSPIDIAG[8]. This latter, a tool developed by the national performance support agency (Agence Nationale d’Appui à la Performance : ANAP), sheds light on a given facility, bringing together data from different databases (PMSI, annual institutional statistics, etc.) in a single tool [2]. The databases contain around eighty indicators which are likely to be filled each year by all the hospitals. In French health system, there are approximately 1600 hospitals classified as public, nonprofit private and commercial private.

All the three newspapers propose a ranking per surgery specialty, for instance a ranking of weight loss surgery. Our analysis in this paper is focused on weight loss surgery. The remarks and comments developed here are valid for all the specialties.

[1] LAMSADE, University Paris Dauphine, email: brice.mayag@dauphine.fr

[2] http://health.usnews.com/best-hospitals

[3] http://www.usnews.com/pubfiles/BH_2014_Methodology_Report_Final_Jul14.pdf

[4] http://www.nhs.uk

[5] http://classement-hopitaux.nouvelobs.com/

[6] http://hopitaux.lepoint.fr/

[7] http://sante.lefigaro.fr

[8] http://hospidiag.atih.sante.fr

2.1 Weight loss surgery

Bariatric surgery[9] (weight loss surgery) includes a variety of procedures performed on people who are obese. Weight loss is achieved by reducing the size of the stomach with a gastric band or through removal of a portion of the stomach (sleeve gastrectomy or biliopancreatic diversion with duodenal switch) or by resecting and rerouting the small intestines to a small stomach pouch (gastric bypass surgery).

To identify the "best" hospitals in weight loss surgery, the newspapers combine a part of the following indicators:

1. (CR_1) *Volume of activity:* it is the number of stays of all patients with respect to the value of care and some homogeneous price.
2. (CR_2) *Activity*: number of procedures performed during one year. "Le Point" supposes that if an hospital has a good score on activity then its teams are more trained and often have good results. This opinion is not totally shared by some other experts who estimate that a good score on the activity of an hospital does not imply necessarily that its teams are best. In this case, one should also investigate if this hospital does not focus on getting grants of the government because in France some grants depend on the activity.
3. (CR_3) *Average Length Of Stay (ALOS)*: a mean calculated by dividing the sum of inpatient days by the number of patients admissions with the same diagnosis-related group classification. A variation in the calculation of ALOS can be to consider only the length of stay during the period under analysis. If an hospital is more organized in terms of resources then its ALOS score should be low.
4. (CR_4) *Notoriety*: Its corresponds to the reputation and attractiveness of the hospital.
   For "the Nouvel Observateur", the attractiveness of the hospital depends on the distance between the hospital and the patient's home. This distance is considered significant if it is more than fifty kms. Its reputation reflects the gradual isolation of patients: the more they come from far away, the more the reputation of the institution is important.
   The notoriety indicator of "Le Point" is a percentage of patients treated in the hospital but living in another French administrative department. More the percentage increases, more the hospital is attractive.
5. (CR_5) *Heaviness*: it is a percentage measuring the level of resources consumed (equipment, staff, ...) in the hospital.
6. (CR_6) *Quality score of French National Authority for Health (HAS)* [10]: It is the score (between ● and ● ● ● ● ●) obtained by the hospital after the accreditation and quality visit made by the experts of HAS.
7. (CR_7) *% of By-Pass*: It is the percentage of surgical procedures using gastric bypass system.
8. (CR_8) *Technicality*: this particular indicator measures the ratio of procedures performed with an efficient technology compared to the same procedures performed with obsolete technology. The higher the percentage is, the more the team is trained in advanced technologies or complex surgeries.

Remark 1. *"Le Nouvel Observateur" use the term activity as a composite indicator of ALOS (CR_3) and volume of activity (CR_1).*

2.2 The 2013 results

The rankings given by "Le Nouvel observateur" [12] take into account, in the same tables, both public and private hospitals. They argue that this logic is in spirit of their readers. In terms of MCDA, this justification of the choice of this set of alternatives appears weak and seems to be only a "marketing argument". Table 1 presents the ranking of only 20 public hospitals (among the first hundred hospitals evaluated) in weight loss surgery published by "Le Nouvel observateur" in 2013. These hospitals are evaluated on five indicators: Volume of activity (CR_1), ALOS (CR_3), % of By-Pass (CR_7), Heaviness (CR_5) and Notoriety (CR_4). In their methodology, they mention that they chose indicators which are most significant in terms of medical innovation, but nothing is said about the concrete selection of such indicators. The last column, F_O, concerns the aggregation function used. Again, nothing is said about this function and how they calculated the score of each hospital. We imagine that it could be a simple weighted sum.

Hospitals	CR_1	CR_3	CR_7	CR_5	CR_4	F_O
Georges-Pompidou	406	5.2	55	77	95	**19.3**
Bichat	203	7.8	75	83	94	**18.9**
Ambroise-Paré	193	6.6	90	83	94	**18.7**
Strasbourg	330	6.2	84	79	45	**18.2**
Nice	351	6.5	94	79	20	**18.1**
Nancy	230	6.9	87	81	76	**17.9**
Louis-Mourier	154	5.0	81	81	27	**17.9**
Pitié-Salpetrière	127	6.0	75	79	92	**17.8**
Laon	299	1.8	0	54	58	**17.7**
Lille	233	6.2	68	83	30	**17.4**
Colmar	192	3.5	97	77	19	**17.4**
Conception	287	3.1	28	63	22	**17.3**
Caen	152	6.7	89	79	63	**17.1**
Toulouse	173	4.3	63	77	87	**17.0**
Antibes	181	5.6	96	77	23	**16.9**
Edouard-Herriot	89	4.9	52	81	38	**16.9**
Havre	115	2.7	78	74	9	**16.5**
Jean-Verdier	116	6.7	44	79	32	**16.4**
Timone adultes	69	5.0	32	81	36	**16.3**
Orleans	131	6.1	69	81	41	**16.4**

Table 1. The best 20 hospitals in Weight loss surgery (2013). Source: "Le Nouvel Observateur" [12]

"Le Point" [13] have analyzed 952 hospitals in their rankings. Just 50, 40, 30, 25 or 20 best hospitals per specialty were published. In Table 2, the ranking published in 2013 concerns the 20 best hospitals in weight loss surgery evaluated on Activity (CR_2), (Notoriety) (CR_4) ; ALOS (CR_3) and Technicality: (CR_8). The last column of the table refers to the scores obtained by using an aggregation function F_P. Like the previous newspaper, nothing is said about this function and nothing about the elaboration of criteria. They only indicate that it is a weighted sum.

Among 1308 hospitals analyzed by the last newspaper, "Le Figaro Magazine" [11], only 830 have been evaluated. The rankings published concern the 10 best hospitals per specialty and per French region. We show in Table 3 some best hospitals in eight regions. The criteria used are: Activity (CR_2) and Quality score of French National Authority for Health (HAS) (CR_6). The ranking is based on

[9] http://en.wikipedia.org/wiki/Bariatric_surgery

[10] French National Authority for Health (HAS) aims to improve quality and safety of healthcare. The objectives are to accredit health care organizations and health professionals, to produce guidelines for health professionals (practices, public health, patient safety), to develop disease management for chronic conditions, to advise decision makers on health technologies (drugs, devices, procedures), and to inform professionals, patients, and the public.

a lexicographic order ($CR_6 \ll CR_2$), but nothing about how these rankings were elaborated.

We are not really surprised if the interesting information for researchers about methodologies used by these three newspaper are poor and not available. Indeed, in France, the sales of newspapers devoted to hospital ranking are often the best of the year. So there exist a real competition between the three organisms. Therefore, each of them has to keep secret its methodology.

Hospitals	CR_2	CR_4	CR_3	CR_8	F_P
Bichat	372	80	7.8	94	**17.84**
Nice	253	19	8.2	95	**17.59**
Nancy	208	60	8	90	**17.37**
Ambroise-Paré	140	85	6.5	96	**17.23**
Colmar	165	14	3.8	99	**17.20**
Caen	167	47	6.7	96	**17.14**
Strasbourg	289	25	6.3	82	**17.13**
Georges-Pompidou	394	80	5.5	56	**17.06**
Lille	247	18	4.8	63	**17.02**
Antibes	156	13	5.5	96	**16.75**
Orleans	167	35	6.7	86	**16.66**
Rouen	237	29	5.1	48	**16.55**
Jean-Verdier	174	40	9.7	82	**16.45**
Conception	332	19	3.8	24	**16.44**
Louis-Mourier	166	51	5.3	86	**16.36**
Poissy/St Germain	192	34	4.1	60	**16.30**
Montpellier	297	25	5.6	33	**16.24**
Toulouse	181	73	4.6	50	**15.94**
Amiens	170	28	3.8	10	**15.63**
Laon	242	23	1.4	0	**15.54**

Table 2. The best 20 hospitals in Weight loss surgery (2013). Source: "Le Point" [13]

Hospitals	CR_2	CR_6
Georges-Pompidou	878	● ● ● ● ●
Bichat	384	● ● ●●
Saint-Louis	285	● ● ●●
Rouen	300	● ● ●●
Laon	277	●●
Lille	271	● ● ●●
Caen	179	●●
Nantes	175	●●
Limoges	103	● ● ●
Rennes	89	●●
Montpellier	353	●●
Nice	263	●●
Orleans	206	● ● ●●
Tours	122	● ● ●
Jean-Mermoz Lyon	312	●●
Sens	140	● ● ●
Nancy	305	●●
Colmar	169	●●
Toulouse	352	● ● ●●
Bordeaux	133	●●

Table 3. The best 20 hospitals in Weight loss surgery (2013). Source: "Le Figaro Magazine" [11]

3 Our propositions

We think that, the *elaboration of hospital ranking* is a practical application where algorithms of MCDA and Machine Learning can be applied. Compared to the newspapers, the academic background of researchers of these two domains can help to better understand this kind of real problem and to propose some valid methodologies. Furthermore, there exists available real data to test these methods and algorithms or to elaborate some benchmarks. Of course, to have a good interpretation of results and indicators, there is a need to work with experts from health systems. Let us give below some suggestions indicating how to proceed.

3.1 Machine learning aspects

In hospital rankings problems, machine learning algorithms can help to determine relevant indicators to use, i.e. to determine which relevant criteria, in each specialty, are needed in the MCDA methodologies. In this case, we can use predictive algorithms like decision tree algorithms.

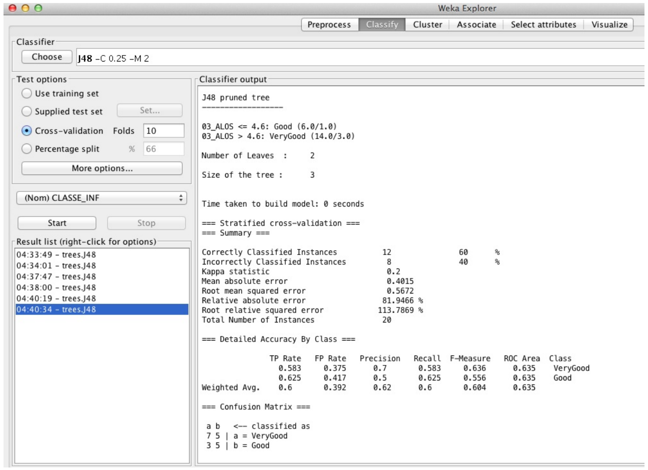
Decision tree learning [9, 15] is one of the most successful techniques for supervised classification learning. It builds classification or regression models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. So the goal is to create a model that predicts the value of a target variable based on several input variables. It is closely related to the fundamental computer science notion of "divide and conquer". A decision node has two or more branches. Leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.

To illustrate our suggestion, let us apply the J48 algorithm of the suite of machine learning software *Weka*[11] to data of hospital rankings given in Tables 1 and 2. J48 is an implementation of the C4.5 algorithm developed by Ross Quinlan [14] to generate a decision tree.

By considering column F_P in Table 2, we can compute two classes from the "Le Point" ranking of weight loss surgery like this: the class VeryGood for hospitals with a score between 16.5 and 18, and the class Good for those having a score between 15 and 16.49. The idea here is to predict these two classes by applying a decision tree algorithm. The Figure 1 shows the results of this example by applying the

[11] Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes. Weka is open source software issued under the GNU General Public License. `http://www.cs.waikato.ac.nz/ml/weka/`.

algorithm J48 of Weka. Only 12 hospitals among 20 have been correctly classified. The decision tree obtained is given by Figure 2. In this classification problem, ALOS seems the only relevant indicator.



Classifier: J48 –C 0.25 –M 2

Test options
○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) CLASSE_INF

Result list (right-click for options)
04:33:49 – trees.J48
04:34:01 – trees.J48
04:37:47 – trees.J48
04:38:00 – trees.J48
04:40:19 – trees.J48
04:40:34 – trees.J48

Classifier output

```
J48 pruned tree
------------------

03_ALOS <= 4.6: Good (6.0/1.0)
03_ALOS > 4.6: VeryGood (14.0/3.0)

Number of Leaves  :     2

Size of the tree :      3


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          12               60      %
Incorrectly Classified Instances         8               40      %
Kappa statistic                          0.2
Mean absolute error                      0.4015
Root mean squared error                  0.5672
Relative absolute error                 81.9466 %
Root relative squared error            113.7869 %
Total Number of Instances               20

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.583     0.375      0.7       0.583     0.636      0.635    VeryGood
                 0.625     0.417      0.5       0.625     0.556      0.635    Good
Weighted Avg.    0.6       0.392      0.62      0.6       0.604      0.635

=== Confusion Matrix ===

 a b   <-- classified as
 7 5 | a = VeryGood
 3 5 | b = Good
```

Figure 1. Applying J48 in Weka from “Le Point” ranking

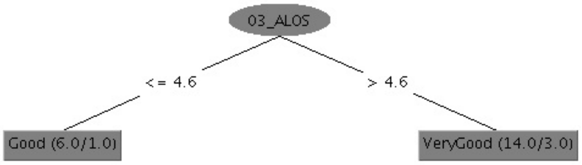


Figure 2. Decision tree from “Le Point” ranking

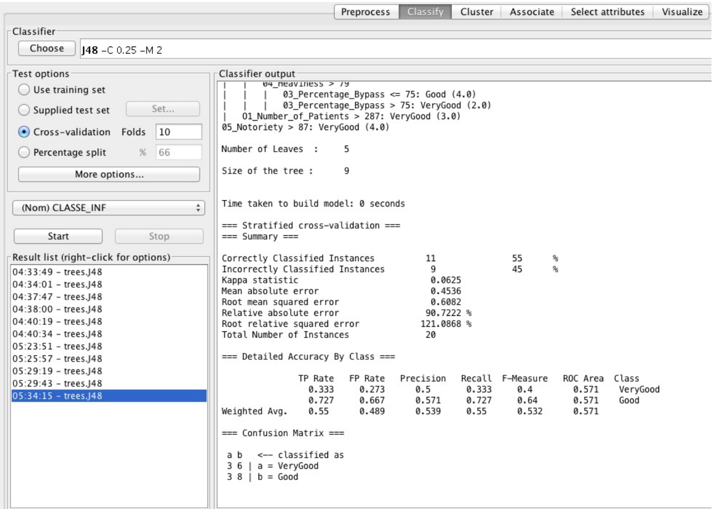
From the “Le Nouvel Observateur” ranking in weight loss surgery (see Table 1), lets us define two classes as follows: the class VeryGood for hospitals with a score belonging to the interval [19.5;17.5], and the class Good for those having a score between 15.5 and 16.49. By applying the algorithm J48 of Weka, Figure 3 shows that only 11 hospitals among 20 have been correctly classified. In the decision tree produced and represented in Figure 4, ALOS seems to be an irrelevant indicator.

3.2 MultiCriteria Decision Aid aspects

As indicated in [3], we have to start with a a number of crucial questions when trying to build an evaluation (ranking) model in MCDA [5, 6]. These questions, known as good practices, are:

1. What is the definition of objects to be evaluated?
2. What is the purpose of the model? Who will use it?
3. How to structure objectives?
4. How to achieve a “consistent family of criteria”?
5. How to take uncertainty, imprecision, and inaccurate definition into account? All the French hospital ranking fail this last point.

After answering these questions, the choice of the suitable MCDA method will be another problem. Some methodologies are based on the weighted sum (e.g. methodologies of “Le Point” and “Le Nouvel Observateur”), because this function is simple and understandable by many persons who are not experts in MCDA.



Classifier: J48 –C 0.25 –M 2

Test options
○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) CLASSE_INF

Result list (right-click for options)
04:33:49 – trees.J48
04:34:01 – trees.J48
04:37:47 – trees.J48
04:38:00 – trees.J48
04:40:19 – trees.J48
04:40:34 – trees.J48
05:23:51 – trees.J48
05:25:57 – trees.J48
05:29:19 – trees.J48
05:29:43 – trees.J48
05:34:15 – trees.J48

Classifier output

```
|   |   04_Heaviness > 79
|   |   |   03_Percentage_Bypass <= 75: Good (4.0)
|   |   |   03_Percentage_Bypass > 75: VeryGood (2.0)
|   01_Number_of_Patients > 287: VeryGood (3.0)
05_Notoriety > 87: VeryGood (4.0)

Number of Leaves  :     5

Size of the tree :      9


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          11               55      %
Incorrectly Classified Instances         9               45      %
Kappa statistic                          0.0625
Mean absolute error                      0.4536
Root mean squared error                  0.6082
Relative absolute error                 90.7222 %
Root relative squared error            121.0868 %
Total Number of Instances               20

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.333     0.273      0.5       0.333     0.4        0.571    VeryGood
                 0.727     0.667      0.571     0.727     0.64       0.571    Good
Weighted Avg.    0.55      0.489      0.539     0.55      0.532      0.571

=== Confusion Matrix ===

 a b   <-- classified as
 3 6 | a = VeryGood
 3 8 | b = Good
```

Figure 3. Applying J48 in Weka from “Le Nouvel Observateur” ranking

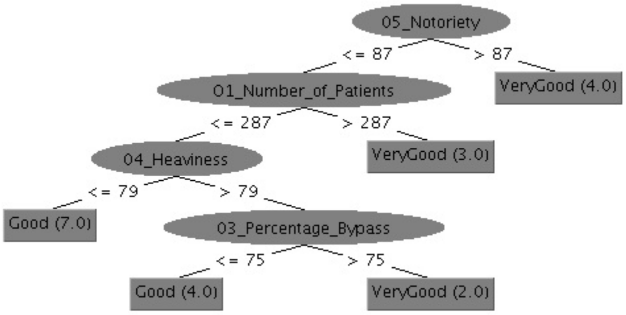


Figure 4. Decision tree from “Le Nouvel Observateur” ranking

If we consider the following four hospitals evaluated on three criteria: Notoriety, ALOS and Technicality:

	Notoriety	ALOS	Technicality
Hospital 1	35	80	90
Hospital 2	37	80	89
Hospital 3	35	40	90
Hospital 4	37	40	89

It seems reasonable to give these preferences: hospital 1 is strictly prefer to the hospital 2 (if ALOS is “weak”, it is preferable to have an hospital with good evaluation in Technicality) and hospital 4 is strictly prefer to hospital 3 (If ALOS is “good”, we prefers in this case an hospital with good evaluation in Notoriety). But it is well known that these aggregation function cannot be model by a weighted sum because they contain some interactions between criteria [4]. Therefore it will be useful to study the dependence between criteria in hospital rankings and then introduce other aggregation functions instead of weighted sum.

We end this section by showing that it is possible to apply an outranking method in this type of application. Because our aim is not to show that the rankings obtained by applying these methods are better than those presented above, we just chose ELECTRE TRI method as an example. ELECTRE TRI [7] is a MCDA method which deals with the sorting problematic. We present hereafter a simple version of ELECTRE TRI, without any preference thresholds and veto, which is sufficient in our context.

Let us denote by $A = \{a_1; a_2; \ldots; a_m\}$ a set of m alternatives or options, $N = \{1; 2; \ldots; n\}$ a set of n criteria or points of view, $C = \{C_1; C_2; \ldots; C_t\}$ a set of ordered categories (C_1 is the worst one and C_t is the best one) and $B = \{b_1; \ldots; b_{t-1}\}$ a set of profiles (reference alternatives which can be fictitious) that separate consecutive categories. Each category C_i, except C_1 and C_t, is limited by two profiles: b_i is the upper limit and b_{i-1} is the lower limit.

The MCDA ELECTRE TRI method assigns alternatives to categories by using the concept of outranking relation $\mathcal{S}$ on $A \times B$. An alternative $a_i \in A$ outranks a profile $b_h \in B$ (denoted $a_i \; \mathcal{S} \; b_h$) if it can be considered at least as good as the latter (i.e., a_i is not worse than b_h), given the values (performances) of a_i and b_h at the n criteria. If a_i is not worse than b_h in every criterion, then it is obvious that $a_i \; \mathcal{S} \; b_h$. However, if there are some criteria where a_i is worse than b_h, then a_i may outrank b_h or not, depending on the relative importance of those criteria and the differences in the evaluations (small differences might be ignored). Roughly speaking,

$$a_i \text{ outranks } b_h \; (a_i \; \mathcal{S} \; b_h) \Leftrightarrow \sum_{1}^{n} k_j \; c_j(a_i, b_h) \geq \lambda.$$

Where

- $c_j(a_i, b_h) = \begin{cases} 1 \text{ if } a_i \; \succsim_j \; b_h \\ 0 \text{ otherwise} \end{cases}$.

  The relation $a_i \; \succsim_j \; b_h$ means that the value of a_i on the criterion j is at least as good as the value of b_h on the same criterion j.
- k_j is the importance (weight) of criterion j such that $\sum_{1}^{n} k_j = 1$.
- λ is the cutting level i.e. a threshold that indicates whether the credibility is significant or not. This parameter is often taken between 0.5 and 1.

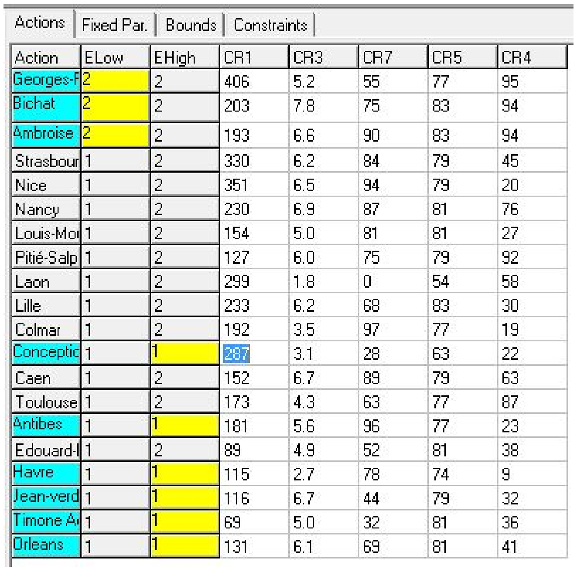
Hence ELECTRE TRI assigns the alternative a_i to the highest category C_h such that a_i outranks b_{h-1} i.e.
for $h = 2, \ldots, t-1$,

$$\begin{cases} a_i \text{ belongs to } C_1 \Leftrightarrow \; \text{not}(a_i \; \mathcal{S} \; b_1) \\ a_i \text{ belongs to } C_h \Leftrightarrow a_i \; \mathcal{S} \; b_{h-1} \text{ and not}(a_i \; \mathcal{S} \; b_h), \\ a_i \text{ belongs to } C_t \Leftrightarrow a_i \; \mathcal{S} \; b_{t-1} \end{cases}$$

We applied ELECTRE TRI on the data given in Tables 1 and 2 by using the software IRIS[12]. This dataset is translated in the performance tables given in Figures 5 and 6.

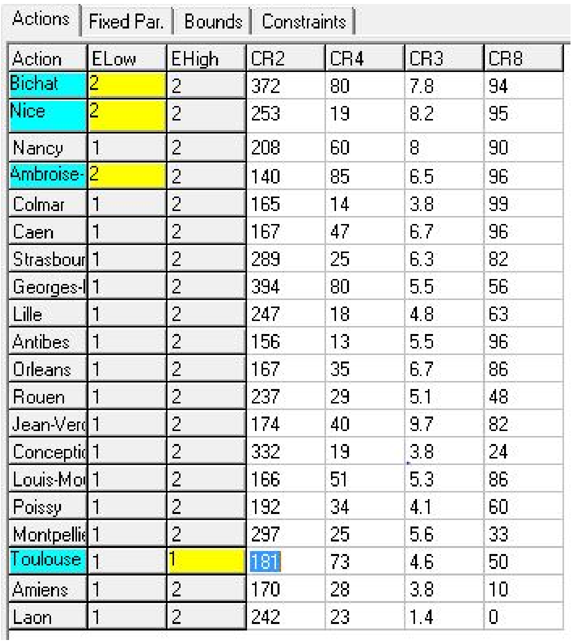
For each problem, we consider two categories C_1 and C_2. The profile between these two categories are presented in Figures 7 and 8. For instance, the profile considered in "Le Point" ranking in weight loss surgery is $b_1 = (150; 60; 5; 80)$. Note that, $g(b_1)$ in Figure 8 corresponds to the values of b_1.

The assignments proposed by ELECTRE TRI is given in Figure 10 and 9 with the values of weights of criteria (denoted by k1, K2,



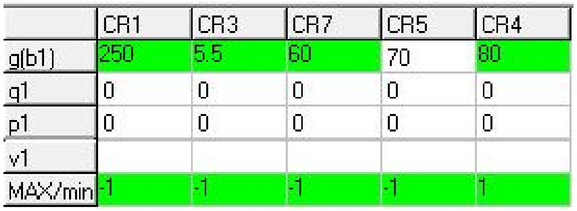
Action	ELow	EHigh	CR1	CR3	CR7	CR5	CR4
Georges-F	2	2	406	5.2	55	77	95
Bichat	2	2	203	7.8	75	83	94
Ambroise	2	2	193	6.6	90	83	94
Strasbour	1	2	330	6.2	84	79	45
Nice	1	2	351	6.5	94	79	20
Nancy	1	2	230	6.9	87	81	76
Louis-Mo	1	2	154	5.0	81	81	27
Pitié-Salp	1	2	127	6.0	75	79	92
Laon	1	2	299	1.8	0	54	58
Lille	1	2	233	6.2	68	83	30
Colmar	1	2	192	3.5	97	77	19
Conceptic	1	1	287	3.1	28	63	22
Caen	1	2	152	6.7	89	79	63
Toulouse	1	2	173	4.3	63	77	87
Antibes	1	1	181	5.6	96	77	23
Edouard-I	1	2	89	4.9	52	81	38
Havre	1	1	115	2.7	78	74	9
Jean-verd	1	1	116	6.7	44	79	32
Timone A	1	1	69	5.0	32	81	36
Orleans	1	1	131	6.1	69	81	41

Figure 5. Performance table of "Le Nouvel Observateur" in weight loss surgery



Action	ELow	EHigh	CR2	CR4	CR3	CR8
Bichat	2	2	372	80	7.8	94
Nice	2	2	253	19	8.2	95
Nancy	1	2	208	60	8	90
Ambroise-	2	2	140	85	6.5	96
Colmar	1	2	165	14	3.8	99
Caen	1	2	167	47	6.7	96
Strasbour	1	2	289	25	6.3	82
Georges-I	1	2	394	80	5.5	56
Lille	1	2	247	18	4.8	63
Antibes	1	2	156	13	5.5	96
Orleans	1	2	167	35	6.7	86
Rouen	1	2	237	29	5.1	48
Jean-Verd	1	2	174	40	9.7	82
Conceptic	1	2	332	19	3.8	24
Louis-Mo	1	2	166	51	5.3	86
Poissy	1	2	192	34	4.1	60
Montpelli	1	2	297	25	5.6	33
Toulouse	1	1	181	73	4.6	50
Amiens	1	2	170	28	3.8	10
Laon	1	2	242	23	1.4	0

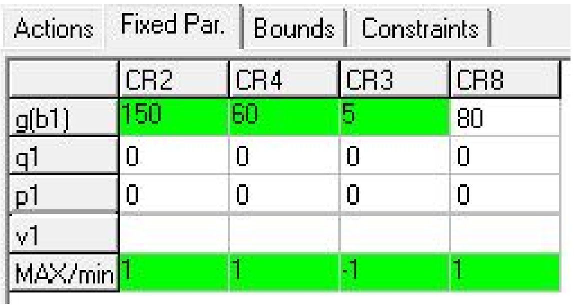
Figure 6. Performance table of "Le Point" in weight loss surgery



	CR1	CR3	CR7	CR5	CR4
g(b1)	250	5.5	60	70	80
q1	0	0	0	0	0
p1	0	0	0	0	0
v1					
MAX/min	-1	-1	-1	-1	1

Figure 7. Profile of "Le Nouvel Observateur" in weight loss surgery

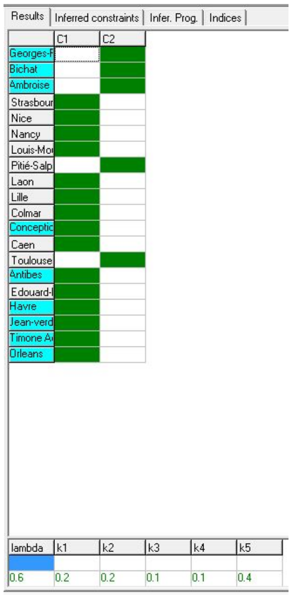
[12] IRIS is a software implementing the ELECTRE TRI method. It is free available at `http://www.lamsade.dauphine.fr/spip.php?rubrique64`



	CR2	CR4	CR3	CR8
g(b1)	150	60	5	80
q1	0	0	0	0
p1	0	0	0	0
v1				
MAX/min	1	1	-1	1

Figure 8. Profile of “Le Point” in weight loss surgery

. . .) and the value of the threshold λ (denoted by lamda). For instance, ELECTRE tri assigns in the same category the five last hospitals whenever you take one of the two rankings given in Tables 2 and 1.



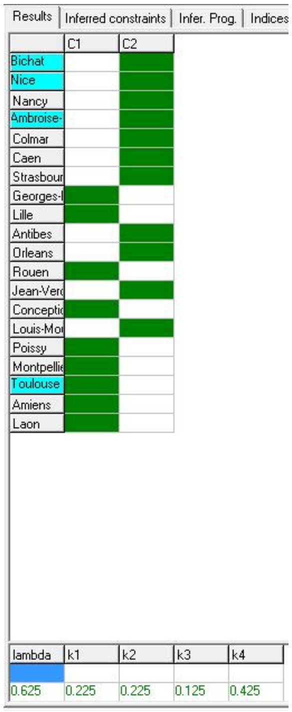
	C1	C2
Georges-F		🟩
Bichat		🟩
Ambroise		🟩
Strasbour	🟩	
Nice	🟩	
Nancy	🟩	
Louis-Mo	🟩	
Pitié-Salp		🟩
Laon	🟩	
Lille	🟩	
Colmar	🟩	
Conceptic	🟩	
Caen	🟩	
Toulouse		🟩
Antibes	🟩	
Edouard-I	🟩	
Havre	🟩	
Jean-verd	🟩	
Timone A	🟩	
Orleans	🟩	

lambda	k1	k2	k3	k4	k5
0.6	0.2	0.2	0.1	0.1	0.4

Figure 9. Assignments of hospitals in “Le Point” ranking related to weight loss surgery

4 Conclusion

We analyzed French hospital rankings, especially in weight loss surgery, made by three newspapers. There is very little official information about how these rankings are made, and the process is not transparent. We showed that this problem is a practical problem where tools of preference learning and MCDA communities (e.g. decision tree and ELECTRE TRI method) can be used in a complementary way.



	C1	C2
Bichat		🟩
Nice		🟩
Nancy		🟩
Ambroise-		🟩
Colmar		🟩
Caen		🟩
Strasbour		🟩
Georges-I	🟩	
Lille	🟩	
Antibes		🟩
Orleans		🟩
Rouen	🟩	
Jean-Verc		🟩
Conceptic	🟩	
Louis-Mo		🟩
Poissy	🟩	
Montpellie	🟩	
Toulouse	🟩	
Amiens	🟩	
Laon	🟩	

lambda	k1	k2	k3	k4
0.625	0.225	0.225	0.125	0.425

Figure 10. Assignments of hospitals in “Le Nouvel Observateur” ranking related to weight loss surgery

REFERENCES

[1] C. A. Bana e Costa and J.-C. Vansnick. The MACBETH approach: basic ideas, software and an application. In N. Meskens and M. Roubens, editors, *Advances in Decision Analysis*, pages 131–157. Kluwer Academic Publishers, 1999.

[2] S. Baron, C. Duclos, and P. Thoreux. Orthopedics coding and funding. *Orthopaedics & Traumatology: Surgery & Research*, 100(1, Supplement):99 – 106, 2014. 2013 Instructional Course Lectures (SoFCOT).

[3] J.-C. Billaut, D. Bouyssou, and P. Vincke. Should you believe in the shanghai ranking? - an mcdm view. *Scientometrics*, 84(1):237–263, 2010.

[4] D. Bouyssou, M. Couceiro, C. Labreuche, J.-L. Marichal, and B. Mayag. Using choquet integral in machine learning: What can mcda bring? In *DA2PL 2012 Workshop: From Multiple Criteria Decision Aid to Preference Learning*, Mons, Belgique, 2012.

[5] D. Bouyssou, Th. Marchant, M. Pirlot, and A. Tsoukiàs. *Evaluation and Decision Models: a critical perspective*. Kluwer Academic, 2000.

[6] D. Bouyssou, Th. Marchant, M. Pirlot, A. Tsoukiàs, and Ph. Vincke. *Evaluation and Decision Models: stepping stones for the analyst*. Springer Verlag, 2006.

[7] J. Figueira, V. Mousseau, and B. Roy. ELECTRE methods. In J. Figueira, S. Greco, and M. Ehrgott, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, pages 133–162. Springer, 2005.

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

[9] M. Hall, I. Witten, and F. Eibe. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers, 2011. 3rd edition. The 2nd edition is free available at `http://home.etf.rs/~vm/os/dmsw/Morgan.Kaufman.Publishers.Weka.2nd.Edition.2005.Elsevier.pdf`.

[10] Ch. Labreuche and F. Le Huédé. Myriad: a tool suite for MCDA. In *Int. Conf. of the Euro Society for Fuzzy Logic and Technology (EUSFLAT)*, pages 204–209, Barcelona, Spain, September 7-9 2005.

[11] J. De Linares. Hôpitaux et cliniques: Le palmarès 2013. *Le Figaro Magazine*, pages 39–49, June 21, 2013. `http://sante.lefigaro.fr/actualite/2013/06/23/20819-palmares-2013-hopitaux-cliniques`.

[12] J. De Linares. Le palmarès national 2013: Hôpitaux et cliniques. *Le*

Nouvel Observateur, pages 77–117, November 28, 2013. http://classement-hopitaux.nouvelobs.com/.

[13] F. Malye and J. Vincent. Hôpitaux et cliniques: Le palmarès 2013. pages 86–142, August 22, 2013. http://hopitaux.lepoint.fr/.

[14] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[15] L. Rokach and O. Maimon. *Data Mining with Decision Trees: Theroy and Applications*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2008.

Session 6

- Invited speaker: "*Scaling Optimization Methods for Data-driven Marketing*", Craig Boutillier, University Toronto, Canada ,

  The emergence of large-scale, data-driven analytics has greatly improved the ability to predict the behavior of, and the effect of marketing actions on, individual consumers. Indeed, the potential for fully personalized "marketing conversations" is very real. Unfortunately, advances in predictive analytics have significantly outpaced the ability of current decision support tools and optimization algorithms, precisely the tools needed to transform these insights into marketing plans, policies and strategies. This is especially true in large marketing organizations, where large numbers of campaigns, business objectives, product groups, etc. place competing demands on marketing resources—the most important of which is customer attention. In this talk, I will describe a new approach, called dynamic segmentation, for solving large-scale marketing optimization problems.
  We formulate the problem as a generalized assignment problem (or other mathematical program) and create aggregate segmentations based on both (statistical) predictive models and campaign-specific and organizational objectives. The resulting compression allows problems involving hundreds of campaigns and millions of customers to be solved optimally in tens of milliseconds. I'll briefly describe how the data-intensive components of the algorithm can be distributed to take advantage of modern cluster-computing frameworks. I will also discuss how the platform supports real-time scenario analysis and re-optimization, allowing decision makers to explore tradeoffs across multiple objectives in real-time.
  Time permitting, I'll hint at how the technique might be extended to solve sequential, stochastic problems formulated as Markov decision processes, and briefly mention other potential applications of this class of techniques.

Session 7

- "*Factorization of large tournaments for the median linear order problem*",
  Alain Guénoche,
  Institut de Mathématiques de Marseille (I2M - CNRS)

- "*Listing the families of Sufficient Coalitions of criteria involved in Sorting procedures*",
  Eda Ersek Uyanik[1], Olivier Sobrie[1,2], Vincent Mousseau[2] and Marc Pirlot[1]
  [1] MATHRO, UMONS, [2] LGI, Ecole Centrale Paris

Factorization of large tournaments for the median linear order problem

Alain Guénoche

Institut de Mathématiques de Marseille (I2M - CNRS)
alain.guenoche@univ-amu.fr

Résumé : *Computing a median linear order for a given set of linear orders on n elements, is an standard task for preference aggregation. This problem is formalized by a tournament (complete directed graph) with n vertices, arcs corresponding to majority preferences. To build a median linear order is to make it transitive, realizing a minimum number of arc-reversal operations. These arcs define the remoteness of any linear order to this tournament. A median linear order has a smallest remoteness value. The computation of a minimum series of arc reversals is usually made using a* Branch & Bound *algorithm which cannot be applied when n is greater than a few tens. In this text we try to decompose a large tournament ($n > 100$) into sub-tournaments and to assemble the median orders on each one into a linear order on n elements. We show, making several simulations on random tournaments, weighted or unweighted, that this decomposition strategy is efficient.*

Mots-clés *: Preferences, linear orders, tournament, median order*

1 Problem

A group E of *experts* ($|E| = m$), ranking a set X of items ($|X| = n$), defines a linear order *profile* $\Pi = \{S_1, S_2, \dots, S_m\}$. Let δ be the symmetric difference distance between linear orders considered as item pair sets on X. We try to establish a linear order π from this profile, being a median order for Π according to distance $\delta : S \times S \to \mathbb{N}$. It means that

$$\sum_{i=1,\dots,m} \delta(S_i, \pi) \qquad (1)$$

is minimum over the linear order set $\mathcal{S}$ on X [2].

To build a median order from profile Π a table T indexed on $X \times X$ is first computed. $T(x, y) = |\{S \in \Pi \text{ such that } x \prec_S y\}|$; evidently $T(x, y)+T(y, x) = m$. This table is associated to a tournament having arc (x, y) directed from x to y iff $T(x, y) > T(y, x)$. This arc can be weighted by $w(x, y) = T(x, y) - T(y, x)$ and $w(y, x) = 0$ in the W array.

Often, in practical problems, preferences are not linear orders because of ties. In that case, preferences are weak orders. Nevertheless, the summarizing of a profile can be done the same way, defining a majority tournament.

The remoteness of any order O to the tournament is defined as the sum of weights of arcs (x, y) such that y is before x in O. These are the *reversal arcs* for O. Obviously, a linear order is equivalent to a transitive tournament. When it is not, a set of arcs to reverse is searched to make the computed tournament transitive. This set must have minimal weight to give a median order. It is the Kemeny problem [11], which is NP-hard (see [10] for a large survey). Using a *Branch & Bound* algorithm, a linear order π with minimum remoteness to the tournament is built. It is a median order for profile Π [8, 3]. Its remoteness W is the sum of weights of the reversal arcs, that is arcs directed from y to x, when x is before y in π :

$$W_\Pi(\pi) = \sum_{x \prec_\pi y} w(y,x). \tag{2}$$

For unweighted tournaments, it is the number of reversal arcs which must be minimized. It becomes the Slater problem [13], which is the same as before with weights all equal to 1. It is also NP-hard and the same algorithm is used to solve both problems.

In this article, we are interested with large problems ($100 \leq n < 1000$). They generally do not occur in preference aggregation, because experts cannot rank such a large number of items. But this type of instance exists when comparing a large number of items evaluated by marks or criteria, as for the Universities of the Shanghai ranking [4], or genes ordered according to their suggested importance in a genetic disease [1], considering many gene expression data. Nevertheless, we keep the preference aggregation scheme to develop our factorization method.

A transitive tournament corresponds to a single linear order, which is easily built ranking the internal half-degrees in increasing order. But if the tournament contains many circuits, the *Branch & Bound* procedure can be very long and fail because of computation time or sufficient memory to extend the tree. Each node corresponds to a beginning section (a prefix) of a linear order which can be extended to a median one [8]. Despite many careful efforts [5, 6], as soon as n is larger than 20 elements, the tree can overpass 500 000 nodes. Then, heuristics are used to get an upper bound to the remoteness of an optimal linear order from the tournament, and also an approximate solution to the problem.

2 Classical heuristics

We only keep two of them, because Borda's method (increasing order of the sum of item ranks in the profile) and the Smith & Payne method [14] (reversal of arcs involved in the largest number of 3-cycles) have been found inefficient for the problem size we tackle.

2.1 The increasing order of internal half-degrees

Vertex x is said to be *dominated* by vertex y when $T(y,x) > T(x,y)$ and the internal half degree of x is the number of vertices dominating x. It is very natural to put at the first place, in the searched linear order, a vertex having the smallest half-degree and to continue according to this increasing order. This simple and fast heuristic is the most efficient for unweighted tournaments, that is, giving frequently the smallest remoteness value among classical heuristics in the following simulated problems. The degree sums are computed in $O(n^2)$ and the increasing order in $O(n \log n)$.

2.2 The greedy heuristic

The greedy heuristic uses the same principle as for the *Branch & Bound* procedure, except the tree of beginning sections is not developed. At each step the item promising the smallest remoteness is selected and the costs of the remaining items are updated. The column sums of the weight table are first computed

$$Sum(x) = \sum_{y \in X} w(y,x). \tag{3}$$

$Sum(x)$ is the contribution of x to the remoteness of an order beginning by x. At each step

- item x such that $Sum(x)$ is minimum is selected ;
- weights are updated : $Sum(y) \leftarrow Sum(y) - w(x,y)$.

This heuristic is clearly in $O(n^2)$; it is the best one for weighted tournaments, in the same sense as before.

There are many other stochastic optimization heuristics, for instances, Variable neighborhood search [9] or Noising methods [7]. We do not consider them in this study, because of parameters to adapt, computation time to manage, or computer codes only made by the authors. But any heuristic solution giving a linear order can be the starting point of optimization procedures. Again, we select only two of them that are deterministic.

2.3 Two local optimization procedures

Any ordering heuristic establishes a linear order $O = (o_1, o_2, \ldots, o_n)$ on X. To improve it, we apply two local optimization procedures.

– The first one is very classical : two consecutive items such that $w(o_{i+1}, o_i) > 0$ are searched. It is clear that transposing these elements will make the remoteness decrease, erasing a reversal arc. This procedure is iteratively repeated until there is no such pair to apply.

– The second one is only for weighted tournaments. For each element o_j, we sarch for the closest item o_i placed before o_j and dominated by o_j. If it exists, we have $o_i \prec_O o_j$, $w(o_j, o_i) > 0$ and (o_j, o_i) is the shortest reversal arc from o_j as it is depicted in Fig.1. It is interesting to swap o_j and o_i if the items placed between o_i and o_j do not create reversal arcs with a larger weight. This is checked by summing values

$$Q = \sum_{i>k>j} w(o_k, o_j) + \sum_{i>k>j-1} w(o_i, o_k). \qquad (4)$$

The first sum corresponds to arcs ending in o_j and the second one to arcs starting from o_i which would became reversal arcs after swapping o_i and o_j. The last weight $w(o_i, o_{j-1})$ is not counted in the second sum because, if it is positive, it suffices to transpose o_{j-1} and o_i, this latter taking the place of o_j.

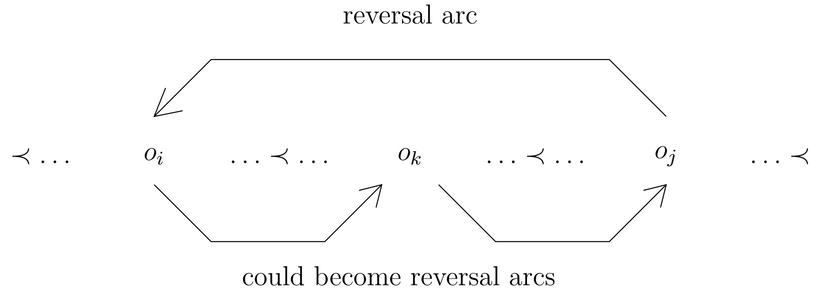


Fig. 1 : The search configuration to apply the second optimization procedure

So, if $w(o_j, o_i) > Q$ the interval $(o_i, o_{i+1}, \ldots, o_{j-1}, o_j)$ becomes either $(o_j, o_{i+1}, \ldots, o_i, o_{j-1})$, or $(o_j, o_{i+1}, \ldots, o_{j-1}, o_i)$ according to $w(o_i, o_{j-1})$ which is positive or not.

As far as I know, this latter optimization procedure is new and its efficiency has been tested. It is fast, since for each element, it suffices to go back to the last dominated item and to apply formula (4) within this interval; its complexity is $O(n^2)$. In the following, let $Best_H$ be the smallest remoteness value given by the heuristics followed by these local optimization procedures.

3 Factorization of a tournament

For median linear order problems of large size ($n \gg 100$), these heuristics are poorly efficient. We study the idea of a tournament decomposition in sub-tournaments, that is to separate the X items into clusters of elements that are close in a median linear order. It could be efficient to compute a linear order for each class and to concatenate them making an order on X. We are going to test if this *composed order* is closer to the tournament, with a remoteness smaller than the classical heuristic ones when they are applied to X as a whole.

3.1 A balanced decomposition

The linear order given by the best heuristic ($BestH$, the one giving the smallest remoteness) easily infers a balanced decomposition. Given a number of clusters p, it suffices to build classes as intervals along this order. May be the items are not optimaly ranked, but one can hope intervals are. In this decomposition, the n/p first ranked items in $BestH$ are in the first class, the next n/p in the second, and so on. One gets a partition in balanced clusters denoted P_B.

3.2 A partition based on a distance

Considering the W table of the arc weights, one can associate to each element x a bipartition : Let $x+$ be the set of items which would be ranked before x because they dominate it, and $x-$ those which would be placed after x because it dominates them.

$$x+ = \{z \in X | w(z,x) > 0\} \text{ and } x- = \{z \in X | w(x,z) > 0\}.$$

Using these bipartitions, on can define a dissimilarity index on X

$$D(x,y) = \Delta(x+,y+) + \Delta(x-,y-) \tag{5}$$

in which $\Delta(x+,y+)$ is the symmetric difference distance between sets $x+$ and $y+$ (resp. $x-$ and $y-$).

Remark : D is not a distance, because $D(x,y) = 0$ if $w(x,y) = w(y,x) = 0$.

Proposition 1 *If T is a transitive tournament,*
- *Two consecutive elements in its median order have distance equal to 2 ;*
- *$D(x,y)$ is proportional to the rank difference between x and y in the median order ;*

Proof
Let $x \prec y$ be two consecutive elements in the median order corresponding to a transitive tournament. Classes $x+$ and $y+$ (resp. $x-$ and $y-$) only differ by a single element, x (resp. y) and so $D(x,y) = 2$. In the same way, if x and y are separated by k items in the order, $D(x,y) = 2(k+1)$. Thus, values increase along rows from the diagonal, and D is a distance (because there is no tie in preferences). This is the definition of a robinsonian distance.

Consequently, homogeneous classes according to D would gather close elements in a median linear order. The number of clusters, implying the average number of items per sub-tournament, will be defined by a simulation process described in section 4.

The partitioning algorithm is based on an optimization criterion. Given a partition of n items in p classes, denoted $P = \{P_1, \dots, P_p\}$, it tends to minimize the sum M of the average distances of each element to the items belonging to its class.

$$M = \sum_{k=1}^{p} [\sum_{x \in P_k} \frac{1}{|P_k|} \sum_{y \in P_k} D(x,y)] \tag{6}$$

The resulting partition P_M is computed by an iterative procedure similar to k-means. One starts from the atomic partition only made with singletons. At each iteration one element is assigned to the class for which its average distance is minimum. It stops when there are p clusters and no more element to transfer.

3.3 Composition, Complexity and Efficiency

For each class from P_B or P_M, one evaluates
- its rank index value, equal to the average of its item ranks in the best heuristic order ;
- the sub-tournament corresponding to this class, with weights given in W ;

– a linear order minimizing, as much as possible, its remoteness to the sub-tournament. For the following computations, I retain the first heuristic for unweighted problems and the second one for weighted tournaments. Median orders can be searched, but the computation time would be much larger even with decomposition in small classes.

Then, the linear orders corresponding to clusters are concatenated according to their rank index values, making in this way a *composed linear order*. The local optimization procedures are applied, making finally two linear orders, $Comp_B, Comp_M$ for the two decomposition methods.

The balanced decomposition algorithm is linear. The distance array computation in in $O(n^3)$, since for each item pair, the relative positions of $n-2$ elements are compared. Partition P_M is established by an iterative algorithm, without knowing its iteration number, as for k-means, which is well known for its efficiency. Then, classical heuristics are applied to each class followed by local optimizations applied to the composed order which remain in $O(n^2)$.

Nevertheless, the *composed linear order* method is fast. For a tournament having 1000 nodes, a P_B linear order in computed in 1”20 and in 19”30 for P_M, using an ordinary desk computer.

4 Simulations and results

We generate two series of random problems.

4.1 Random permutations profiles

Selecting m random permutations of order n [12] makes a profile Π and a W matrix. The two classical heuristics give the $BestH$ linear order. Fixing the number of classes p makes on one side, the partitioning P_B and the composed linear order $Comp_B$ and on the other side, calculating distance D and applying the partitioning algorithm gives partition P_M and the $Comp_M$ linear order. For these three orders their remoteness to the tournament is measured.

Tests are made on 100 profiles with the same parameters. Each row in Table 1 gives the average remoteness. The three first columns are for unweighted tournaments and the three last are for weighted ones.

n	m	p	$BestH$	$Comp_B$	$Comp_M$	$BestH$	$Comp_B$	$Comp_M$
100	10	3	805	788	**784**	755	731	**719**
100	20	3	832	814	**812**	1236	1203	**1187**
100	30	3	844	827	**825**	1569	1534	**1522**
200	30	4	3584	3520	**3514**	6825	6678	**6614**
200	50	5	3621	3542	**3536**	9047	8820	**8784**
200	100	6	3645	**3554**	3560	13129	**12762**	12782
500	100	5	23726	23476	**23456**	86600	85336	**85107**
500	100	10	23726	**23313**	23502	86600	**84670**	85635
500	100	15	23726	**23294**	23555	86600	**84636**	86078

Table 1 : Remoteness values of the orders given by heuristics on unweighted (left) and weighted (right) tournaments

The composed linear orders are much better than the best classical heuristic. They win at each trial, except for a few problems with $n = 100$. But these are average results and, for a specific problem, both decomposition methods must be applied. Two questions remain : which is the optimal number of classes for factorization and how far are these figures from the optimum (a median linear order) ?

To answer the first one, we consider 100 orders on 300 items ($n = 300, m = 100$), for which we seek the optimal number of classes in the average. In Table 2, the two first columns correspond to unweighted tournament and the two others are again for weighted ones. Classical heuristics give remoteness values independent of p, respectively 8333 and 30240, always larger than those obtained by factorizing the tournament.

p	$Comp_B$	$Comp_M$	$Comp_B$	$Comp_M$
4	8232	8222	29786	29644
5	8205	8197	29659	**29538**
6	8183	**8192**	29551	29561
⋮	⋮	⋮	⋮	⋮
10	8144	8225	29409	29832
11	8138	8226	**29397**	29856
12	**8137**	8243	29405	29913
13	8140	8245	29419	29938

Table 2 : Average remoteness values of the orders given by decomposition of unweighted (left) and weighted (right) tournaments, making the number of classes p vary.

Remoteness average values at first decrease when the class number increases, then they increase. It is why we don't go further. The minimum value is obtained with class number depending on the method. But the best decomposition is reached for $p = 11$ or 12 corresponding to around 30 elements per class.

4.2 Tournaments with bounded remoteness

Selecting independent random permutations as before generates tournaments far from transitive and the computed orders have a large remoteness. The best linear order cannot be seen as a consensus order, because there is no meaningful consensus to these profiles. So now, we are going to generate tournaments from a unique linear order, making transpositions between random items. Let t be the parameter counting the transposition number. There are two generating processes :

- Starting from the natural linear order, corresponding to a transitive unweighted tournament, t pairs are selected at random ; when $x < y$, $T(x, y) = 1$ and $T(y, x) = 0$. Transposing (x, y) will make $T(x, y) = 0$ and $T(y, x) = 1$. Doing so, we are sure there exists a linear order with a remoteness to the final tournament lower than or equal to t.
- The m permutations are built from the natural order transposing t random pairs in each one. The weighted tournament is then computed, according to the usual majority rule. But if t is small compared to the possible number of pairs, the consensus and median order would be the natural order.

The first tests are for unweighted tournaments with $n = 300$ are given in Table 3, in which 2000, 3000, 4000 random transpositions are made, over the 44850 possible ones. So the median order must be very close to the natural order, for which the remoteness is also calculated. The same algorithms as before are run with a factorization in 10 clusters :

n	t	p	$BestH$	$Comp_B$	$Comp_M$	$NatOrd$
300	2000	10	1965	1903	**1898**	1895
300	3000	10	2915	2796	**2786**	2779
300	4000	10	3841	3661	**3643**	3628

Table 3 : Average remoteness values for the $BestH$ heuristic, the two factorization methods and the natural order expected to be a median one, on unweighted tournaments obtained after t random transpositions.

These are again average values over 100 problems. The given bound, equal to t is lightly improved by the $BestH$ heuristic, but much more by the composed linear orders. And partition P_M provides values very close to those of the natural order suspected to be optimal.

The second test is made with permutations on which 100 pair transpositions have been made before to calculate the weighted tournament. Factorizations are always made with 10 classes.

n	m	t	p	$BestH$	$Comp_B$	$Comp_M$	$NatOrd$
300	30	100	10	2065	1721	**1434**	1555
300	50	100	10	800	677	**558**	544
300	100	100	10	56	51	**49**	46

Table 4 : Average remoteness values for the same linear orders as in Table 3, on weighted tournaments obtained after t random transpositions on m natural orders.

The larger is the number of permutations (m), the lower is the remoteness because the corresponding tournament becomes more and more transitive. As it can be seen in Table 4, the second decomposition method proves its efficiency for problems with a strong consensus.

5 Conclusion

For a large tournament, the factorization strategy is always the winner in these experiments. And so, it is better to concatenate small orders optimized from sub-tournaments than to compute an optimized linear order from the whole tournament. More, for tournaments close to be transitive, the P_M decomposition gives linear orders very close to the median one.

So, for a large specific tournament coming from real data, I will first determine an optimal number of classes with the balanced partitioning, which is very fast and compute the distance array between items. Then, around this class number, I will try the partitioning algorithm. A computer program, in C, can be required to the author. A last trial with a 1000 vertices tournament provides, with the balanced decomposition in 15 clusters, the smallest remoteness value it founds.

Références

[1] S. Aerts, D. Lambrechts, Y. Moreau et. al. Gene prioritization through genomic data fusion, *Nature Biotechnology*, 24, 5, pp. 537-544, 2006.

[2] J.P. Barthélemy, B. Monjardet. The Median Procedure in Cluster Analysis and Social Choice Theory, *Mathematical Social Sciences*, 1, pp. 235-267, 1981.

[3] J.P. Barthélemy, A. Guénoche, O. Hudry. Median linear orders : Heuristics and Branch and Bound algorithm. *European Journal of Operational Research*, 42, pp. 555-579, 1989.

[4] J.C. Billaut, D. Bouyssou, P. Vincke. Should you believe in the Shanghai ranking ? An MCDM view. *Scientometrics*, 84 (1), 237-263, 2010, DOI : 10.1007/s11192-009-0115-x.

[5] I. Charon, A. Guénoche, O. Hudry, F. Woirgard. A Bonsaï Branch and Bound method applied to voting theory, *Proceedings of "Ordinal and Symbolic Data Analysis"*, OSDA'95, E. Diday et al. (Eds.), Springer Verlag, 309-318, 1996.

[6] I. Charon, O. Hudry, F. Woirgard. Ordres médians et ordres de Slater des tournois, *Mathématiques et Sciences Humaines*, 133, pp. 23-56, 1996.

[7] I. Charon, O. Hudry. The noising methods : a generalization of some metaheuristics, *European Journal of Operational Research*, 135 (1), 86-101, 2001.

[8] A. Guénoche. Un algorithme pour pallier l'effet Condorcet, *R.A.I.R.O. Recherche Opérationnelle*, 11, 1, 77-83, 1977.

[9] P. Hansen, N. Mladenović. Variable neighborhood search : Principles and Applications, *European Journal of Operational Research*, 130(3), 449-467, 2001.

[10] O. Hudry. On the computation of median linear orders, of median complete preorders and of median weak orders, *Mathematical Social Sciences*, 64, 2-10, 2012.

[11] J.G. Kemeny. Mathematics without numbers, *Daedalus*, 88, 577-591, 1959.

[12] A. Nijenhuis, H. Wilf. *Combinatorial Algorithms*, Academic Press, New-York, 1975.

[13] P. Slater. Inconsistencies in a schedule of paired comparisons, *Biometrika*, 48, 303-312, 1961.

[14] A.F.M. Smith, C.D. Payne. An algorithm for determining Slater's i and all nearest adjoining orders, *British Journal of Mathematical and Statistical Psychology*, 27, 49-52, 1974.

Listing the families of Sufficient Coalitions of criteria involved in Sorting procedures

Eda Ersek Uyanık[1,4], Olivier Sobrie[2,3,4], Vincent Mousseau[3] and **Marc Pirlot[4]**

Abstract. Certain sorting procedures derived from ELECTRE TRI such as MR-Sort or the Non-Compensatory Sorting (NCS model) model rely on a rule of the type: if an object is better than a profile on a "sufficient coalition" of criteria, this object is assigned to a category above this profile. In some cases the strength a coalition can be numerically represented by the sum of weights attached to the criteria and a coalition is sufficient if its strength passes some threshold. This is the type of rule used in the MR-Sort method. In more general models such as Capacitive-MR-Sort or NCS model, criteria are allowed to interact and a capacity is needed to model the strength of a coalition. In this contribution, we want to investigate the gap of expressivity between the two models. In this view, we explicitly generate a list of all possible families of sufficient coalitions for a number of criteria up to 6. We also categorize them according to the degree of additivity of a capacity that can model their strength. Our goal is twofold: being able to draw a sorting rule at random and having at disposal examples in view of supporting a theoretical investigation of the families of sufficient coalitions.

1 Introduction

A *sorting method*, in Multiple Criteria Decision Analysis, is a procedure for assigning objects (or alternatives) described by their evaluation on several criteria to ordered categories. ELECTRE TRI [17, 10] is a sorting method based on an outranking relation. Basically, each category has a lower limit profile which is also the upper limit profile of the category below. An object is assigned to a category if it outranks the lower limit profile of this category but does not outrank its upper limit profile. MR-Sort is a simple version of ELECTRE TRI. MR-Sort assigns an object to a category if its evaluations are better than the value of the lower limit profiles on a majority of criteria and this condition is not fulfilled with respect to the upper limit profile of the category. More precisely, a weight w_i is attached to each criterion $i = 1, 2, \dots, n$ and the object $a = (a_1, a_2, \dots, a_n)$ is assigned to a category above profile $b = (b_1, b_2, \dots, b_n)$ whenever the sum of the weights of the criteria for which $a_i \geq b_i$ passes some threshold λ. Otherwise, it is assigned to a category below b.

An intermediary sorting method in between ELECTRE TRI and MR-Sort was proposed and characterized by Bouyssou and Marchant [1, 2]. It is known as the *Non Compensatory Sorting* (NCS) model. Consider the simple case in which there are only two categories (e.g. good vs. bad) and no veto. In such a case, an object is assigned to the category "good" if it is better than the lower limit profile of this category on a *sufficient coalition of criteria*. How do they define the "sufficient coalitions of criteria"? Basically, these can be any collection of criteria with the following property: a coalition that contains a sufficient coalition of criteria is itself sufficient.

We claimed that MR-Sort is a particular case of a NCS model. Indeed, with MR-Sort, a set of criteria is a sufficient coalition iff the sum of the weights of the criteria in the set is at least as large as the threshold λ. To fix the ideas consider the following example. A student has to take 4 exams to be admitted in a school. To be successful, he has to take a mark of at least twelve (over twenty) in each of these exams, with at most one exception. In this case the lower limit profile of the category "succeed" is the vector $(12, 12, 12, 12)$ and the sufficient coalitions of criteria are all subsets of at least 3 subjects for which the student's mark is at least 12. Denote the student's marks by $a = (a_1, a_2, a_3, a_4)$. The sufficient coalitions can be represented by associating a weight to each course, e.g. each exam receives a weight equal to $1/4$, and choosing an appropriate threshold, here $3/4$. The assignment rule then reads: x succeeds iff $|\{i : x_i \geq 12\}| \times 1/4 \geq 3/4$, which is indeed the typical form of a MR-Sort rule.

Not all assignment rules based on sufficient coalitions can be represented by additive weights and a threshold. For instance, assume that the exams subjects are French language (1), English language (2), Mathematics (3) and Physics (4). To be successful, a student has to take at least 12 points in one of the first two and in one of the last two. If the weights of the four subjects are respectively denoted w_1, w_2, w_3, w_4 and the threshold is λ and if we want to represent the rule using these weights and threshold, we see that these parameters have to fulfill the following inequalities:

$$\begin{cases} w_1 + w_3 & \geq & \lambda \\ w_1 + w_4 & \geq & \lambda \\ w_2 + w_3 & \geq & \lambda \\ w_2 + w_4 & \geq & \lambda \\ w_1 + w_2 & < & \lambda \\ w_3 + w_4 & < & \lambda \end{cases}$$

These conditions are contradictory. Indeed, summing up the first four inequalities, we get that $\lambda \leq 1/2 \sum_{i=1}^{4} w_i$, while

[1] email: eda.uyanik@gmail.com

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

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

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

summing up the last two yields $\lambda > 1/2 \sum_{i=1}^{4} w_i$.

Our goal with this paper is to investigate the gap of expressivity between MR-Sort and NCS model (without veto). In this perspective, we analyze the possible families of sufficient coalitions up to a number of criteria equal to 6. We start by listing all these families, which raises difficulties due to the combinatorial and complex character of this issue. Then we study which families of sufficient coalitions are representable by an inequality involving weights attached to the criteria, as in MR-Sort. We partition the set of all families of sufficient coalitions according to the type of inequality they fulfill. All these families are counted and listed. This study aims first at an explicit description of the families of sufficient criteria, up to $n = 6$, in order to support further more theoretical investigations and practical applications. As a by-product, it also enables to make simulations by drawing at random a MR-Sort model or a NCS model. This proves useful e.g. for testing the efficiency of algorithms designed for learning a NCS model on the basis of assignment examples.

The rest of the paper is organized as follows. In Section 2, we state the problem more formally, we introduce the notion of capacity and we recall combinatorial results related to the enumeration of families of sufficient coalitions. Section 3 describes how the sets of sufficient coalitions were generated. In Section 4, we explain how we partitioned the families of sufficient coalitions; the size of each class of this partition is computed. The next section explains how these results can be exploited for simulation purposes and a short conclusion follows.

2 Background

2.1 Numerical representation of the sufficient coalitions

In MR-Sort, the set of sufficient coalitions of criteria can be represented numerically. In other words it is possible to check whether a set of criteria is sufficient by checking whether an inequality is satisfied. More precisely, there is a family of nonnegative weights $w_1, w_2, \dots, w_n$ and a nonnegative threshold λ such that a set of criteria $A \subseteq \{1, 2, \dots, n\}$ is sufficient iff

$$\sum_{i \in A} w_i \geq \lambda. \tag{1}$$

We assume w.l.o.g. that $\sum_{i=1}^{n} w_i = 1$. Such a representation is generally not unique. For instance, in the example above involving 4 criteria, the family of sufficient coalitions is formed by all subsets of at least 3 criteria; this family can be represented by assigning equal weights to all criteria and using threshold value 3/4. Alternatively, one could use e.g. $w_1 = .2, w_2 = .2, w_3 = .3, w_4 = .3$ as weights and $\lambda = .70$ as threshold to represent the same family of coalitions.

We saw also above that, in general, not all families of sufficient coalitions can be specified by an inequality such as (1). If this is not the case, is there another kind of inequality that can be used? Actually, any family of sufficient coalitions can be represented using a *capacity* μ and a threshold λ. We briefly recall what is a capacity. A capacity is a set function $\mu : 2^n \to \mathbb{R}_+$ which is monotone w.r.t. to set inclusion, i.e. for all $A, B \subseteq \{1, 2, \dots, n\}, A \subseteq B \Rightarrow \mu(A) \leq \mu(B)$ (monotonicity) and $\mu(\emptyset) = 0$. We impose w.l.o.g. that $\mu(\{1, 2, \dots, n\}) = 1$ (normalization). Note that a capacity is not additive, in general, which means that it does not necessarily satisfy the property: $\mu(A \cup B) = \mu(A) + \mu(B)$ whenever $A \cap B = \emptyset$. If it does, then the capacity μ is said to be additive and it is a probability. This means that there are weights $w_1, w_2, \dots, w_n$ such that $\mu(A) = \sum_{i \in A} w_i$, for all set $A \subseteq \{1, 2, \dots, n\}$. A (non necessarily additive) capacity can be given by means of an interaction function (or Möbius transform) m. One has, for all $A \subseteq \{1, 2, \dots, n\}$:

$$\mu(A) = \sum_{B \subseteq A} m(B) \tag{2}$$

where m is a set function $2^n \to \mathbb{R}$ which satisfies $\sum_{B \subseteq \{1,2,\dots,n\}} m(B) = 1$ and $\sum_{B: i \in B \subseteq A} m(B) \geq 0$, for all $i \in \{1, 2, \dots, n\}$ and $A \subseteq \{1, 2, \dots, n\}$. The capacity defined by (2) is a probability iff $m(B) = 0$ whenever $|B| > 1$. A capacity is said to be k-additive when k is the largest cardinality of the subsets for which m is different from 0. Probabilities are 1-additive (or simply "additive") capacities.

Proposition 1 *Any family of sufficient coalitions can be represented as the set of subsets $A \subseteq \{1, 2, \dots, n\}$ verifying*

$$\mu(A) \geq \lambda, \tag{3}$$

for some capacity μ and threshold $\lambda \geq 0$. Conversely, if μ is a capacity and λ is a nonnegative number, the set of subsets A satisfying the inequality $\mu(A) \geq \lambda$ is a family of sufficient coalitions.

Proof. A family of sufficient coalitions is a family of subsets such that any subset containing a subset of the family is itself in the family. Define a nonnegative set function μ letting $\mu(A) = 1$ if A is a sufficient coalition and 0 otherwise. One can see that μ is monotone, and therefore a capacity, due to the characteristic properties of the families of sufficient coalitions. It is also normalized. Define the threshold $\lambda = .5$. Clearly $\mu(A) \geq .5$ iff A is a sufficient coalition. The proof of the converse is also straightforward.

As a consequence of this result, in a NCS model, the set of sufficient coalitions can be either listed or specified by an inequality such as (3). In a preference learning perspective, the latter representation may be at an advantage since it opens the perspective of using powerful optimization techniques (see [13] for the learning of a NCS model on this basis)[5]. As already observed in the case of weights, the capacity and threshold used for representing a family of SC are generally not unique.

In the sequel we will be interested in parsimonious representations, i.e. representations of a family of SC as the set of coalitions A satisfying (3), using a k-additive capacity, with k as small as possible. The smaller k, the smaller the number of

[5] In [13], the NCS model without veto is called *capacitive MR-Sort model*. Both models are essentially equivalent

parameters to identify capacity μ, for instance in a learning process. If $k = 1$, the family of SC can be represented by an inequality of type (1), which involves determining the value of $n + 1$ parameters (the weights w_i and the threshold λ). If a family of SC is representable using a 2-additive capacity, then we can write $\mu(A) = \sum_{i \in A} m_i + \sum_{i,j \in A, i \neq j} m_{ij}$, where we abuse notation denoting $m(\{i\})$ by m_i and $m(\{i, j\})$ by m_{ij}. In this case, learning μ requires the determination of $\frac{n(n+1)}{2} + 1$ parameters.

2.2 Minimal sufficient coalitions

The set of SC may be large (typically exponential in n), but one can avoid listing them all. A *minimal sufficient coalition* (MSC) is a SC which is not properly included in another SC. Knowing the set of MSC allows to determine all SC since a coalition is sufficient as soon as it contains a MSC. A family of MSC is any collection of subsets of $\{1, 2, \ldots, n\}$ such that none of them is included in another. In other words, a set of MSC is an *antichain* in the set of subsets of $\{1, 2, \ldots, n\}$ (partially) ordered by inclusion. It is well-known that the number of antichains in the power set of $\{1, 2, \ldots, n\}$ is $D(n)$, the nth Dedekind number ([15], sequence A000372). These numbers grow extremely rapidly with n and no exact closed form is known for them. These numbers have been computed up to $n = 8$; these values appear in Table 1.

n	$D(n)$
0	2
1	3
2	6
3	20
4	168
5	7581
6	7828354
7	2414682040998
8	56130437228687557907788

Table 1. Known values of the Dedekind numbers $D(n)$

Remark. The Dedekind numbers are also the number of monotone (more precisely, positive [5]) Boolean functions in n variables. Clearly, the set of sufficient coalitions can be represented as the set of n-dimensional Boolean vectors which give the value 1 to a monotone Boolean function, and conversely. Another application of the Dedekind numbers is in game theory. They are the numbers of simple games with n players in minimal winning form [16, 6].

One way of simplifying the study of the families of sufficient coalitions consists in keeping only one representative of each class of equivalent families of SC. Two families will be considered as equivalent, or isomorphic, if they can be transformed one into the other just by permuting the labels of the criteria. Consider e.g. the following family of minimal SC for $n = 4$: $\{2, 4\}, \{2, 3\}, \{1, 3, 4\}$. It consists of 2 subsets of 2 criteria and one of 3 criteria. There are 12 equivalent families that can be obtained from this one, by permuting the criteria labels (the criterion which does not show up in the set of 3 criteria can be chosen in 4 different ways and the two criteria which distinguish the two pairs can be chosen in 3 different ways). The number $R(n)$ of *inequivalent* families of SC is known for $n = 0$ to $n = 7$ ([15], sequence A003182). $R(7)$ was only recently computed by Stephen and Yusun [14]. Table 2 lists the known values of $R(n)$.

n	0	1	2	3	4	5	6	7
$R(n)$	2	3	5	10	30	210	16353	490013148

Table 2. Number of inequivalent families of sufficient coalitions of n criteria

Finally we recall Sperner's theorem ([4], p.116-118), a result that will be useful in the process of generating all possible families of SC. The maximal size of an antichain in the power set of a set of n elements is $\binom{n}{\lfloor n/2 \rfloor}$. Hence the latter is the maximal number of sets in a family of minimal SC.

3 Listing the families of minimal sufficient coalitions

For generating all families of MSC and selecting a representative of each class of equivalent families, we follow a strategy similar to the one used in [14]. We describe it briefly. The families of MSC can be partitioned according to their *type* (called "profile" in [14]). The type of a family of MSC is an integer vector $(k_1, k_2, \ldots, k_n)$, where k_i represents the number of coalitions of i criteria in the family. For instance, the family $\{2, 4\}, \{2, 3\}, \{1, 3, 4\}$, for $n = 4$, is of the type $(0, 2, 1, 0)$, since it involves two coalitions of 2 criteria and one of 3 criteria. For any feasible type, $\sum_{i=1}^{n} k_i \leq \binom{n}{\lfloor n/2 \rfloor}$, due to Sperner's theorem.

The listing algorithm roughly proceeds as follows:

1. generate all type vectors $(k_1, k_2, \ldots, k_n)$ in lexicographic increasing order;
2. for each type, generate all families of subsets of $\{1, 2, \ldots, n\}$ having the right type and eliminate those that are not antichains, i.e. those involving a subset that is included in another subset;
3. for each type and for each family of this type, the list of remaining families is screened for detecting families that are equivalent, counting them and eliminating them from the list of families of the type considered.

This algorithm outputs a list containing a representative of each class of equivalent families of MSC for each type.

Example. For $n = 3$, the inequivalent families of MSC, with their number of equivalent versions, are displayed in Table 3.

Remarks:

1. there exist two additional families which do not appear in Table 3:
   - the empty family, corresponding to the case in which no coalition is sufficient, which means, for a sorting procedure, that all objects are assigned to the "bad" category;
   - the family of which the sole element is the empty set; this means that all coalitions are sufficient, even the empty

Type	Representative	# equivalent
(1,0,0)	{{1}}	3
(2,0,0)	{{1}, {2}}	3
(3,0,0)	{{1}, {2}, {3}}	1
(0,1,0)	{{1,2}}	3
(1,1,0)	{{1}, {2, 3}}	3
(0,2,0)	{{1, 3}, {2, 3}}	3
(0,3,0)	{{1, 2}, {1, 3}, {2, 3}}	1
(0,0,1)	{{1,2,3}}	1
Total	8	18

Table 3. Number of inequivalent families of minimal sufficient coalitions

one, and consequently, all objects are sorted in the "good" category.

Adding these two extreme cases to the counts in the last line of Table 3 yields values that are consistent with Tables 2 and 1.

2. for $n = 3$, every possible class type has a single representative. For larger values of n, this is no longer the case. For instance, for $n = 4$, we have 3 inequivalent representatives for type $(0, 3, 0, 0)$:

Type	Representative	# equivalent
(0,3,0,0)	{{1, 3}, {1, 2}, {3, 4}}	12
(0,3,0,0)	{{2, 4}, {1, 2}, {1, 4}}	4
(0,3,0,0)	{{2, 4}, {3, 4}, {1, 4}}	4

These three inequivalent families are the three sorts of non-isomorphic 3-edge graphs on 4 vertices.

3. in the sequel, in the absence of ambiguity, we shall drop the brackets around the coalitions and the commas separating the elements of a coalition in order to simplify the description of a family of SC; for instance, the first family of type (0,3,0,0) above will be denoted by : $\{13, 12, 34\}$ instead of $\{\{1, 3\}, \{1, 2\}, \{3, 4\}\}$.

The algorithm sketched above can be made more efficient by implementing the following properties (see [14], lemma 2.4 for a proof) linking the families of MSC.

1. There is a one-to-one correspondence between families consisting exclusively of k_i MSC of cardinality i and families consisting exclusively of $\binom{n}{i} - k_i$ MSC of cardinality i. In other terms, there is a bijection between the families of the type $(0, \ldots, 0, k_i, 0, \ldots, 0)$ and these of the type $(0, \ldots, 0, \binom{n}{i} - k_i, 0, \ldots, 0)$. For instance, in Table 3, generating family $\{12\}$ of type (0,1,0), automatically yields family $\{13, 23\}$ of type (0,2,0). The number of representatives in both types are identical (three, in the latter example).
2. If a family of MSC on n criteria contains at least one singleton, then the remaining MSC of the family do not involve this criterion and hence belong to a type of family of MSC on $n - 1$ criteria. In the example of $n = 3$, knowing the families of MSC on 2 criteria allows to generate the families on three criteria for which one criterion alone is a sufficient coalition. For instance, if criterion 1 alone is sufficient, one can build all families in which 1 is a MSC by putting together with 1 each family of MSC on criteria 2 and 3, i.e.: {}, {2},{3},{2, 3} and {23}. This, however, will not allow to directly compute the number of representatives of each type, since some families, involving more than one singleton as MSC, can be generated in several ways. For instance, $\{1, 2\}$ will be obtained both when starting from the singleton 1 as a MSC and completing this family by MSC included in $\{2, 3\}$, and, starting from the singleton 2 and completing this family by MSC extracted from $\{1, 3\}$.
3. There is a one-to-one correspondence between families of MSC belonging to type $(k_1, k_2, \ldots, k_{n-1}, 0)$ and these belonging to the "reverse" type $(k_{n-1}, \ldots, k_2, k_1, 0)$. For instance, starting from the family $\{1, 2\}$ belonging to type (2,0,0) and taking the complement of each MSC, one obtains the family $\{23, 13\}$, which belongs to (0,2,0). This correspondence allows to halve the computations for $D(n)$ and $R(n)$.

Using this algorithm on a cluster, we have computed the list of all inequivalent families of MSC for $n = 2$ to $n = 6$. The results, grouped by type, are available at `http://olivier.sobrie.be/shared/mbfs/`. For illustrative purposes, the case $n = 4$ is in Appendix A.

4 Partitioning the families of sufficient coalitions

4.1 Checking representability by a k-additive capacity

Our main goal in this section is to partition the set of families of MSC, for fixed n, in categories $\mathcal{C}_k$, which are defined as follows.

Definition 1 *A family of sufficient coalitions belongs to class $\mathcal{C}_k$ if*

1. *it is the set of all subsets A of $\{1, 2, \ldots, n\}$ satisfying an inequality of the type: $\mu(A) \geq \lambda$, where μ is a normalized k-additive capacity and λ a non-negative real number;*
2. *k is the smallest integer for which such an inequality is valid.*

It is clear that all equivalent families of MSC belong to the same class $\mathcal{C}_k$. Hence it is sufficient to check for *one* representative of each class of equivalent families of MSC whether or not it belongs to $\mathcal{C}_k$.

The checking strategy is the following. For each inequivalent family of MSC (listed as explained in Section 3), we iteratively check whether it belongs to class $\mathcal{C}_k$, starting from $k = 1$ and incrementing k until the test is positive (we know, by proposition 1, that this will occur at the latest for $k = n$). The test can be formulated as a linear program. Basically, we have to write constraints imposing that $\mu(A) \geq \lambda$ for each sufficient coalition A and that the same inequality is not satisfied for all other coalitions , which will be called *insufficient* coalitions. It is enough to write these sorts of constraints only for the minimal sufficient coalitions and for the maximal insufficient coalitions. The set of minimal sufficient (resp. maximal insufficient) coalitions will be denoted SCMin (resp. SIMax).

To formulate the problem as a linear program, we use formula (2), which expresses the value of the capacity μ as a linear combination of its associated interaction function m. This enables to control the parameter k which fixes the k-additivity of the capacity. When checking whether a family of MSC belongs to class $\mathcal{C}_k$, we set the values of the variables $m(B)$ to 0 for all sets B of cardinality superior to k; the remaining values of the interaction function are the main variables in the linear program. The following constitutes the general scheme of the linear programs used for each class $\mathcal{C}_k$:

$$\left\{\begin{array}{rcll} & \max\ \varepsilon & & \\ \mu(A) & \geq & \lambda & \forall A \in \text{SCMin} \\ \mu(A) & \leq & \lambda - \varepsilon & \forall A \in \text{SIMax} \\ \mu(A) & = & \sum_{B \subseteq A} m(B) & \forall A \in \text{SCMin} \cup \text{SIMax} \\ \sum_{B: i \in B \subseteq A} m(B) & \geq & 0 & \forall i \in \{1, 2, \ldots, n\} \\ & & \text{and} & \forall A \subseteq \{1, 2, \ldots, n\} \\ \sum_{B \subseteq \{1,2,\ldots,n\}} m(B) & = & 1 & \\ \lambda, \varepsilon & \geq & 0 & \end{array}\right. \tag{4}$$

Note that the variables $m(B)$ are not necessarily positive (except for $|B| = 1$). To fix the ideas, we show how to instantiate the third, fourth and fifth constraints in the cases $k = 1$ and $k = 2$.

- $k = 1$: 1-additive capacity
  - . $\mu(A) = \sum_{i \in A} m_i,\ \forall A \in \text{SCMin} \cup \text{SIMax}$
  - . $m_i \geq 0,\ \forall i \in \{1, 2, \ldots, n\}$
  - . $\sum_{i \in \{1,2,\ldots,n\}} m_i = 1,$

  where m_i stands for $m(\{i\})$
- $k = 2$: 2-additive capacity
  - . $\mu(A) = \sum_{i \in A} m_i + \sum_{i,j \in A,\ i \neq j} m_{ij},\ \forall A \in \text{SCMin} \cup \text{SIMax}$
  - . $m_i + \sum_{j \in A,\ j \neq i} m_{ij} \geq 0,\ \forall i \in \{1, 2, \ldots, n\}$ and $\forall A \ni i,\ A \subseteq \{1, 2, \ldots, n\}$
  - . $\sum_{i \in \{1,2,\ldots,n\}} m_i + \sum_{i,j \in \{1,2,\ldots,n\},\ i \neq j} m_{ij} = 1,$

  where m_i stands for $m(\{i\})$ and m_{ij} for $m(\{i, j\})$.

Writing the constraints for the 3-additive case requires the introduction of a third family of variables m_{ijl} for each subset $\{i, j, l\}$ of three different criteria (in addition to the already introduced variables m_i and m_{ij}).

4.2 Results

For $n = 1$ to 6 and for each family in the list of inequivalent families of MSC, we checked whether this family belongs to $\mathcal{C}_k$, starting with $k = 1$ and incrementing its value until the check is positive. The results are presented in Table 4 for the number and proportion of inequivalent families in classes $\mathcal{C}_2$ and $\mathcal{C}_3$. The families that are not in these classes belong to class $\mathcal{C}_1$. Up to $n = 6$, inclusively, there are no families in classes $\mathcal{C}_4$ or above, which means that all families can be represented using a 3-additive capacity (in the worst case). Up to $n = 5$, inclusively, 2-additive capacities are sufficient. Table 5 represents a similar information but each family in the list of inequivalent families is weighted by the size of the equivalence class it represents. In other words, this is the result that would have been obtained by checking all families of MSC for belonging to class $\mathcal{C}_1$, $\mathcal{C}_2$ or $\mathcal{C}_3$.

n	$R(n)$	$\mathcal{C}_2$	$\mathcal{C}_3$
0	2	0 (00.00 %)	0 (00.00 %)
1	3	0 (00.00 %)	0 (00.00 %)
2	5	0 (00.00 %)	0 (00.00 %)
3	10	0 (00.00 %)	0 (00.00 %)
4	30	3 (10.00 %)	0 (00.00 %)
5	210	91 (43.33 %)	0 (00.00 %)
6	16 353	15 240 (93.19 %)	338 (02.07 %)

Table 4. Number and proportion of inequivalent families of MSC that are representable by a 2- or 3-additive capacity

n	$D(n)$	$\mathcal{C}_2$	$\mathcal{C}_3$
0	2	0 (00.00 %)	0 (00.00 %)
1	3	0 (00.00 %)	0 (00.00 %)
2	6	0 (00.00 %)	0 (00.00 %)
3	20	0 (00.00 %)	0 (00.00 %)
4	168	18 (10.71 %)	0 (00.00 %)
5	7 581	4 294 (56.64 %)	0 (00.00 %)
6	7 828 354	7 584 196 (96.88 %)	145 502 (01.86 %)

Table 5. Number and proportion of all families of MSC that are representable by a 2- or 3-additive capacity

The information displayed in Table 4 (resp. 5) is represented in graphical form in Figure 1 (resp. 2). The cases of 0, 1 and 2 criteria are not represented since all families can be represented by a 1-additive capacity. These figures clearly show that the proportion of families that can be represented by means of a 1-additive capacity, i.e. by additive weights, decreases quite rapidly with the number of criteria. In contrast, the proportion of families that can be represented by a 2-additive capacity grows up to more than 93% from $n = 3$ to $n = 6$. The proportions slightly differ depending on whether only inequivalent families or all families are taken into account. One can observe that the proportion of families in class $\mathcal{C}_2$ is a bit larger when considering all families (Table 5 and Figure 2).

Examples. As a matter of illustration, we describe a few examples for $n = 4$ and $n = 6$. The list of all inequivalent MSC for $n = 5$, which are not representable by a 1-additive capacity, is in appendix B. The categorization in classes $\mathcal{C}_k$ is available at `http://olivier.sobrie.be/shared/mbfs/` for $n = 4, 5, 6$.

1. Here are the three families of MSC on 4 criteria that cannot be represented using a 1-additive capacity (they can be by a 2-additive capacity).

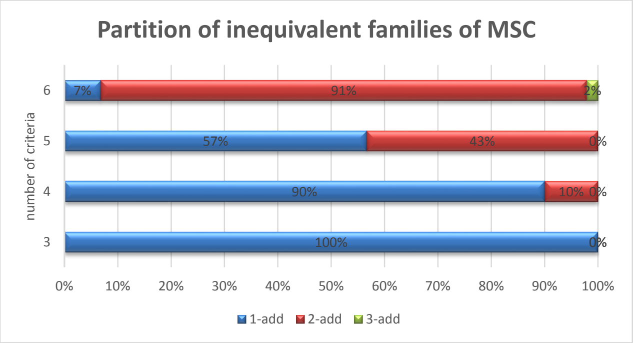


Figure 1. Proportion of inequivalent families of MSC in classes $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$

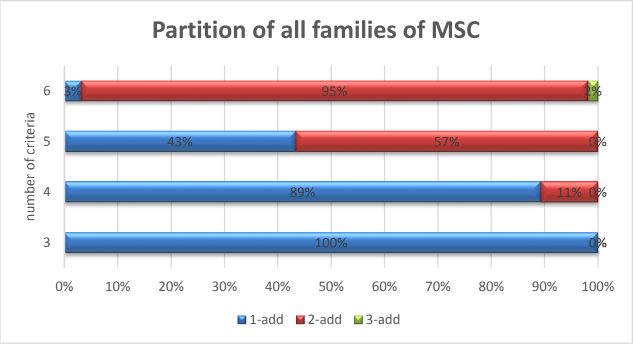


Figure 2. Proportion of all families of MSC in classes $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$

Type	Representative	# equivalent
(0,2,0,0)	{23, 14}	3
(0,3,0,0)	{13, 12, 34}	12
(0,4,0,0)	{13, 14, 23, 24}	3

These three inequivalent families yield, by permutations of the criteria labels, a total of 18 families that can only be represented using a 2-additive capacity.

The last inequivalent family is precisely the example that we used in Section 1 to show that not all families of SC can be represented by a 1-additive capacity. In contrast, it can be represented, for instance, by setting $m_1 = m_2 = m_3 = m_4 = 1/6$ and $m_{13} = m_{14} = m_{23} = m_{24} = 1/12$, while the other pairwise interactions m_{12} and m_{34} are set to 0. We then have: $\mu(13) = \mu(14) = \mu(23) = \mu(24) = 5/12$ while $\mu(12) = \mu(34) = 4/12$. Setting the threshold λ to $9/24$ allows to separate the sufficient coalitions from the insufficient. This representation is by no means unique. We construct another capacity by setting $m_1 = m_2 = m_3 = m_4 = 1/3$, $m_{12} = m_{34} = -1/6$ and $m_{13} = m_{14} = m_{23} = m_{24} = 0$. We have: $\mu(13) = \mu(14) = \mu(23) = \mu(24) = 2/3$ while $\mu(12) = \mu(34) = 1/2$. Setting the threshold λ to $7/12$ also separates the sufficient from the insufficient coalitions.

Note that the second example, a family of type (0,3,0,0) already appeared in Remark 2 after Table 3. In this case all inequivalent families of the type (0,3,0,0) belong to the same class $\mathcal{C}_2$ (see Appendix A). This is not always the case. For $n = 5$, there are for instance six different inequivalent families of the type (0,0,4,0,0), four of which are representable by a 2-additive capacity (and not by a 1-additive capacity; see Appendix B), while the remaining two are representable by a 1-additive capacity. There are many more examples with 5 criteria.

Note also that the first and the last example are *complementary* in the sense of the first property allowing to speed up the enumeration of the families of MSC described at the end of Section 3. Both these families are composed of pairs of criteria; the two pairs in the first family are disjoint from the four in the third family and all pairs are either in one or the other family. In such a situation, it is clear that both families belong to the same class $\mathcal{C}_k$.

2. Here are two examples of inequivalent families of MSC on 6 criteria that are not representable by a 2-additive capacity but require a 3-additive capacity. There are 338 such inequivalent families which yield, through permutations, a total of 145 502 families[6]. A simple example is of the type (0,0,4,0,0,0). The MSC are $\{136, 234, 125, 456\}$. There are 30 equivalent families that can be derived from this family by permutation. Another, much more complex example is of the type (0,1,7,1,0,0). The MSC are $\{135, 256, 345, 36, 234, 456, 1245, 146, 123\}$. There are 360 families that are equivalent to this one through permutations.

In the 338 families, no MSC consists of a single criterion;

[6] If all permutations of the criteria labels were yielding different families, the total number of families would be 338 × 720 = 243 360

none of them involves 5 criteria. The largest number of MSC in a family is 16, the maximal cardinality of a family of MSC on 6 criteria being the Sperner number 20.

5 Usefulness of this analysis

5.1 Applications

The above results, although limited to 6 criteria, maybe useful for different purposes, mainly related to the choice of a sorting model and to simulation.

5.1.1 Choice of a sorting model

In the introduction, we argued that the MR-Sort model might not be sufficiently flexible to accommodate certain assignment rules of interest. The quick decrease with n (illustrated by Figures 1 and 2) of the proportion of rules that can be represented by an inequality comparing a sum of weights to a threshold (corresponding to families of MSC in class $\mathcal{C}_1$) shows that it may indeed be useful to consider more general rules. For $n = 4$, only 18 rules in a total of 168 cannot be represented by a 1-additive capacity. For 5 criteria, there is no need to consider more complicated models than these using a 2-additive capacity. And for $n = 6$, in most of the cases (93% in terms of inequivalent families of MSC and more than 96% if we consider all families of MSC), a 2-additive capacity is enough. These considerations are important in the case one wants to learn a Capacitive-MR-Sort model (i.e. a NCS model without veto) as in [13]. Knowing the minimal value of k enabling to represent the set of MSC on n criteria allows to limit the number of parameters (the interaction function m) that have to be elicited or learned on the basis of examples.

Obviously, in many applications, the number of criteria may exceed 6 and it would therefore be useful to extend the analysis for $n > 6$. Using the same methods as we did, it could be possible to solve the case $n = 7$. But from $n = 8$ on, methods based on enumeration become impracticable: the number $R(8)$ of inequivalent families of MSC is not even known. Alternative approaches would consist in trying to find bounds on the cardinal of the classes $\mathcal{C}_k$ or to obtain characterizations of the families in the different classes and use these to generate examples, whenever they exist, in the various categories.

5.1.2 Simulation

Recently, methods have been proposed to learn variants of the ELECTRE TRI sorting model on the basis of assignment examples [7, 18, 12, 13]. It has also been done [8] for a ranking method based on *reference points* proposed by Rolland [9, 3]. Consider e.g. a learning algorithm designed to learn a MR-Sort model, as in [12]. Real data sets can be used to test the performance of the algorithm. But for learning algorithms which aim at selecting a rule in a specific family of sorting rules, it is also needed to perform the following test, with artificial data. When a set of assignment examples is generated by a known MR-Sort model, we would like to verify that the algorithm, when applied to these examples, learns a model similar to the original one. If some noise is added to the learning set, one expects that the algorithm remains robust. In order to design such tests, we have to draw at random a MR-Sort model, i.e. the profiles, the criteria weights and a threshold. Drawing the profiles and the threshold at random does not raise major problems. An algorithm for drawing weights summing up to 1 in a uniform way is also well-known [11].

In order to perform the same type of tests in the case of the Capacitive-MR-Sort model (or the NCS model without veto), we are facing a difficulty. How can one draw at random a capacity, or more particularly a k-additive capacity? How can one define a uniform distribution on the set of capacities? On second thought, we moved to another formulation of this question. What we have to do is to draw at random, uniformly (in some sense), a MR-Sort rule or a Capacitive-MR-Sort rule, not a capacity. And this makes a difference, since the representation of a Capacitive-MR-Sort rule by an inequality involving a capacity and a threshold is not unique (as observed previously), hence there is a representation bias in this way of proceeding. Note that this remark also applies to drawing at random an MR-Sort model. The alternative is thus to *select a rule at random*, i.e. a family of MSC. That's what our results allow to do, up to $n = 6$. There is no need to test the algorithm for several equivalent versions of the same rule (i.e. for families of MSC that only differ by a permutation of the criteria labels). We can thus sample the set of inequivalent families (each weighted proportionally to the size of its equivalence class). To draw a rule uniformly at random from the set of all Capacitive-MR-Sort rules on n criteria (for $n \leq 6$), proceed as follows:

1. prepare a file in which all inequivalent families of MSC on *criteria* are listed together with the size of their equivalence class; let y_l denote the lth family and s_l the size of its equivalence class, for $l = 1, \dots R(n)$;
2. scan this list and sequentially assign to each family y_l an interval of s_l consecutive integer numbers: y_l is assigned the interval $[N_l, N_l + s_l - 1]$, where $N_l = \sum_{j=1}^{l-1} s_j + 1$;
3. draw uniformly at random an integer number N between 1 and $N_{R(n)}$;
4. find l such that N belongs to the interval $[N_l, N_l + s_l - 1]$ and retrieve the representative of the family of MSC that occupies the lth position in the list.

Note that the lists of inequivalent families also permit to consider non-uniform distributions and to draw at random from them according to an arbitrary probability distribution on the families.

6 Conclusion

In this work, we explored the families of minimal sufficient coalitions as they appear in sorting models such as MR-Sort and Capacitive-MR-Sort. This exploration is limited to small numbers of criteria because of the huge number of such models. Our goal was at least twofold:

1. to have at disposal and make generally available a detailed picture of the possible families of sufficient coalitions for

as large as possible numbers of criteria; this information could help further investigations related in particular to the characterization of the families of sufficient coalitions that can be separated from the insufficient ones by an inequality involving a k-additive capacity.

2. to have at disposal and make generally available a list of the possible sorting rules in the NCS model, in order to enable to draw a rule at random according to any specified probability distribution and use it in simulations. The space needed to store these lists and the time to scan them can be reduced, at least somewhat, by retaining only inequivalent rules.

Further efforts in the future could lead to obtain the list of inequivalent families of sufficient coalitions for $n = 7$. Another interesting topic is the theoretical study of the different classes $\mathcal{C}_k$. Alternatively, other approaches to subdividing the set of all families of sufficient coalitions could be of practical and theoretical interest.

REFERENCES

[1] 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)

[2] 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)

[3] Bouyssou, D., Marchant, T.: Multiattribute preference models with reference points. European Journal of Operational Research 229(2), 470 – 481 (2013)

[4] Caspard, N., Leclerc, B., Monjardet, B.: Finite Ordered Sets : Concepts, results and uses. Encyclopedia of Mathematics and its applications, Cambridge University Press (2012)

[5] Crama, Y., Hammer, P.L.: Boolean Functions - Theory, Algorithms, and Applications, Encyclopedia of mathematics and its applications, vol. 142. Cambridge University Press (2011)

[6] Kurz, S., Tautenhahn, N.: On Dedekind´s problem for complete simple games. International Journal of Game Theory 42(2), 411–437 (2013)

[7] 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)

[8] Liu, J., Mousseau, V., Ouerdane, W.: Preference elicitation from inconsistent pairwise comparisons for multi-criteria ranking with multiple reference points. In: Proceedings of the 14th International Conference on Informatics and Semiotics in Organisations (ICISO 14). pp. 120–132. Stockholm (Sweden) (2013)

[9] Rolland, A.: Reference-based preferences aggregation procedures in multi-criteria decision making. European Journal of Operational Research 225(3), 479 – 486 (2013)

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

[11] Smith, N.A., Tromble, R.W.: Sampling uniformly from the unit simplex (August 2004), Department of Computer Science, Center for Language and Speech Processing, Johns Hopkins University

[12] Sobrie, O., Mousseau, V., Pirlot, M.: Learning a majority rule model from large sets of assignment examples. In: Perny, P., Pirlot, M., Tsoukiás, A. (eds.) Algorithmic Decision Theory. pp. 336–350. Springer (2013)

[13] Sobrie, O., Mousseau, V., Pirlot, M.: Learning the parameters of a majority rule sorting model taking attribute interactions into account. Working paper, University of Mons, Belgium and Ecole Centrale Paris, France (2014), submitted for presentation at DA2PL 2014 Workshop, November 20-21, 2014, Paris, France

[14] Stephen, T., Yusun, T.: Counting inequivalent monotone boolean functions. Discrete Applied Mathematics 167, 15 – 24 (2014)

[15] The OEIS Foundation Inc.: On-line encyclopedia of integer sequences. `https://oeis.org`, [Online; accessed September 13, 2014]

[16] von Neumann, J., Morgenstern, O.: Theory of Games and Economic Behavior. Princeton University Press, NJ, USA, 3rd edn. (1972)

[17] 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)

[18] Zheng, J.: Preference Elicitation for reference based aggregation models: Algorithms and Procedures. Phd thesis, Ecole Centrale Paris (2012)

Appendix

A List of inequivalent families of MSC for $n = 4$

The families are grouped by type. There are 25 possible types, 29 inequivalent families of MSC (plus the trivial case in which all coalitions are sufficient) and 167 families of MSC (plus the same trivial case). Each inequivalent family in the list is associated the size of its equivalence class. All inequivalent families, except three of them, can be represented by a 1-additive capacity. The three other families can be represented by a 2-additive capacity. They are marked in the last column by $\mathcal{C}_2$.

Type	Family of MSC	# eq.	$\mathcal{C}_k$
(0,0,0,0)	{}	1	
(0,0,0,1)	{1234}	1	
(0,0,1,0)	{124}	4	
(0,0,2,0)	{234, 124}	6	
(0,0,3,0)	{134, 123, 124}	4	
(0,0,4,0)	{134, 123, 234, 124}	1	
(0,1,0,0)	{24}	6	
(0,1,1,0)	{14, 123}	12	
(0,1,2,0)	{24, 134, 123}	6	
(0,2,0,0)	{12, 23}	12	
	{23, 14}	3	$\mathcal{C}_2$
(0,2,1,0)	{24, 134, 23}	12	
(0,3,0,0)	{13, 12, 34}	12	$\mathcal{C}_2$
	{24, 12, 14}	4	
	{24, 34, 14}	4	
(0,3,1,0)	{13, 34, 23, 124}	4	
(0,4,0,0)	{24, 12, 13, 34}	3	$\mathcal{C}_2$
	{24, 12, 14, 23}	12	
(0,5,0,0)	{24, 12, 14, 13, 34}	6	
(0,6,0,0)	{24, 12, 14, 34, 23, 13}	1	
(1,0,0,0)	{1}	4	
(1,0,1,0)	{234, 1}	4	
(1,1,0,0)	{14, 2}	12	
(1,2,0,0)	{13, 34, 2}	12	
(1,3,0,0)	{24, 34, 23, 1}	4	
(2,0,0,0)	{4, 3}	6	
(2,1,0,0)	{4, 23, 1}	6	
(3,0,0,0)	{4, 2, 1}	4	
(4,0,0,0)	{4, 2, 3, 1}	1	

B List of inequivalent families of MSC of class $\mathcal{C}_2$ for $n = 5$

We list below the 91 inequivalent families of MSC that cannot be represented by a 1-additive capacity. They can all be represented using a 2-additive capacity. The families are grouped by type. Each inequivalent family in the list is associated the size of its equivalence class.

Type	Family of MSC	# eq.
(0,0,2,0,0)	{135, 234}	15
(0,0,2,1,0)	{234, 125, 1345}	15
(0,0,3,0,0)	{145, 123, 345}	30
	{235, 234, 125}	60
(0,0,3,1,0)	{134, 135, 2345, 124}	60
(0,0,4,0,0)	{145, 234, 345, 124}	15
	{135, 245, 234, 125}	60
	{235, 145, 135, 123}	60
	{134, 345, 234, 125}	10
(0,0,4,1,0)	{245, 123, 234, 125, 1345}	15
(0,0,5,0,0)	{235, 134, 135, 345, 125}	60
	{235, 134, 135, 245, 124}	12
	{235, 145, 134, 245, 124}	60
	{145, 134, 123, 234, 125}	60
(0,0,6,0,0)	{135, 235, 234, 125, 145, 123}	15
	{135, 345, 234, 125, 245, 123}	10
	{345, 235, 234, 125, 124, 134}	60
	{135, 345, 235, 125, 124, 145}	60
(0,0,7,0,0)	{345, 234, 125, 145, 134, 245, 123}	30
	{135, 235, 125, 124, 145, 134, 245}	60
(0,0,8,0,0)	{135, 345, 234, 125, 124, 145, 245, 123}	15
(0,1,1,0,0)	{123, 45}	10
(0,1,2,0,0)	{15, 123, 345}	60
	{12, 134, 345}	60
(0,1,3,0,0)	{235, 14, 123, 125}	60
	{13, 235, 145, 124}	60
	{235, 14, 123, 245}	60
	{24, 134, 135, 123}	30
(0,1,4,0,0)	{235, 15, 245, 123, 234}	120
	{135, 123, 25, 345, 124}	60
	{235, 34, 145, 125, 124}	60
	{24, 235, 135, 123, 125}	20
(0,1,5,0,0)	{345, 235, 15, 234, 134, 123}	30
	{235, 125, 124, 145, 34, 123}	60
	{24, 135, 345, 235, 125, 123}	60
(0,1,6,0,0)	{24, 135, 345, 235, 145, 134, 123}	60
(0,2,0,0,0)	{34, 15}	15
(0,2,1,0,0)	{12, 35, 234}	60
	{145, 23, 25}	60
(0,2,2,0,0)	{24, 13, 125, 345}	30
	{24, 12, 135, 345}	30
	{134, 23, 35, 124}	60
	{13, 12, 245, 234}	120
	{12, 245, 35, 234}	60
(0,2,3,0,0)	{15, 23, 134, 345, 124}	60
	{45, 134, 135, 234, 25}	120
	{135, 123, 45, 125, 14}	60
	{24, 235, 14, 345, 135}	30
	{24, 34, 135, 123, 125}	60
(0,2,4,0,0)	{135, 235, 14, 234, 123, 45}	60
	{14, 35, 234, 125, 245, 123}	15
	{24, 135, 235, 125, 34, 123}	30

Type	Family of MSC	# eq.
(0,3,0,0,0)	{12, 14, 45}	60
	{12, 34, 45}	30
(0,3,1,0,0)	{24, 145, 23, 25}	60
	{34, 14, 35, 125}	60
	{34, 245, 23, 14}	120
	{34, 14, 123, 25}	60
(0,3,2,0,0)	{15, 14, 123, 25, 345}	60
	{24, 12, 134, 35, 145}	30
	{13, 23, 245, 125, 14}	120
	{15, 45, 123, 234, 25}	60
(0,3,3,0,0)	{24, 135, 145, 134, 23, 25}	20
	{12, 35, 234, 145, 13, 245}	60
(0,4,0,0,0)	{34, 15, 14, 35}	15
	{24, 15, 23, 25}	60
	{24, 34, 15, 23}	10
	{24, 34, 15, 35}	60
(0,4,1,0,0)	{13, 34, 35, 25, 145}	60
	{24, 13, 15, 25, 345}	60
	{13, 15, 23, 25, 345}	30
	{34, 14, 45, 125, 23}	60
(0,4,2,0,0)	{24, 12, 35, 145, 134, 23}	60
	{24, 35, 145, 34, 25, 123}	15
(0,5,0,0,0)	{24, 13, 15, 23, 14}	60
	{24, 12, 15, 35, 25}	60
	{24, 12, 15, 35, 34}	12
	{12, 15, 34, 25, 45}	60
(0,5,1,0,0)	{135, 12, 14, 34, 23, 25}	60
	{15, 35, 124, 23, 13, 45}	60
(0,6,0,0,0)	{24, 12, 23, 25, 13, 45}	15
	{24, 12, 35, 34, 25, 13}	10
	{24, 12, 34, 23, 13, 45}	60
	{15, 14, 34, 23, 25, 45}	60
(0,6,1,0,0)	{24, 12, 35, 145, 34, 25, 13}	10
(0,7,0,0,0)	{12, 14, 34, 23, 25, 13, 45}	30
	{24, 12, 15, 14, 35, 34, 45}	60
(0,8,0,0,0)	{24, 12, 15, 34, 23, 25, 13, 45}	15
(1,2,0,0,0)	{34, 15, 2}	15
(1,3,0,0,0)	{24, 15, 3, 25}	60
(1,4,0,0,0)	{13, 2, 14, 35, 45}	15

Session 8

- Invited speaker: "*Surrogate loss functions for preference learning*",
  Krzysztof Dembczynski,
  Poznan University of Technology, Poland,
  In preference learning we use a variety of different performance measures to train and test prediction models. The most popular measures are pairwise disagreement (also referred to as rank loss), discounted cumulative gain, average precision, and expected reciprocal rank. Unfortunately, these measures are usually neither convex nor differentiable, so their optimization becomes a hard computational problem. However, instead of optimizing them directly we can reformulate the problem and use surrogate or proxy loss functions which are easier to minimize. A natural question arises whether optimization of a surrogate loss provides a near-optimal solution for a given performance measure. For some of the performance measures the answer is positive, but in the general case the answer is rather negative. During the tutorial we will discuss several results obtained so far.

Poster session

- “*An Arrow-like theorem over median algebras*”,
  Miguel Couceiro[1] and Bruno Teheux[2],
  1 LAMSADE, Université Paris-Dauphine,
  2 Université du Luxembourg

- “*A Metaheuristic Approach for Preference Learning in Multi-Criteria Ranking based on Reference Points*”,
  Jinyan Liu, Wassila Ouerdane, Vincent Mousseau,
  LGI, Ecole Centrale Paris

- “*Inferring the parameters of a majority rule sorting model with vetoes on large datasets*”,
  Alexandru-Liviu Olteanu, Patrick Meyer,
  Telecom Bretagne

- “*A Dataset Repository for Benchmark in MCDA*”,
  Antoine Rolland and Thi-Minh-Thuy Tran,
  Lab. ERIC, Université Lyon 2

- “*User Experience Driven Design of MCDA Problems with DecisionCloud*”,
  Michel Zam[1,2], Meltem Ozturk[2] and Brice Mayag[2],
  1 KarmicSoft Research,
  2 LAMSADE, Université Paris-Dauphine

An Arrow-like theorem over median algebras

Miguel Couceiro[1] and **Bruno Teheux** [2]

Abstract. We present an Arrow-like theorem for aggregation functions over convervative median algebras. In doing so, we give a characterization of conservative median algebras by means of forbidden substructures and by providing their representation as chains.

1 Introduction and preliminaries

Informally, an aggregation function $f : \mathbf{A}^n \to \mathbf{B}$ is a mapping that preserves the structure of $\mathbf{A}$ into $\mathbf{B}$. Usually, $\mathbf{B}$ is taken equal to $\mathbf{A}$ and is equipped with a partial order so that aggregation functions are thought of as order-preserving maps [7]. In this paper, we are interested in aggregation functions $f : \mathbf{A}^n \to \mathbf{A}$ that satisfy the functional equation

$$f(\mathbf{m}(\mathbf{x}, \mathbf{y}, \mathbf{z})) \;=\; \mathbf{m}(f(\mathbf{x}), f(\mathbf{y}), f(\mathbf{z})), \qquad (1.1)$$

where $\mathbf{A} = \langle A, \mathbf{m}\rangle$ is a *median algebra*, that is, an algebra with a single ternary operation $\mathbf{m}$, called a *median function*, that satisfies the equations

$$\begin{gathered}\mathbf{m}(x, x, y) = x,\\ \mathbf{m}(x, y, z) = \mathbf{m}(y, x, z) = \mathbf{m}(y, z, x),\\ \mathbf{m}(\mathbf{m}(x, y, z), t, u) = \mathbf{m}(x, \mathbf{m}(y, t, u), \mathbf{m}(z, t, u)),\end{gathered}$$

and that is extended to $\mathbf{A}^n$ componentwise. In particular, every median algebra satisfies the equation

$$\mathbf{m}(x, y, \mathbf{m}(x, y, z)) \;=\; \mathbf{m}(x, y, z). \qquad (1.2)$$

An example of median function is the term function

$$\mathbf{m}(x, y, z) = (x \wedge y) \vee (x \wedge z) \vee (z \wedge y) \qquad (1.3)$$

over a distributive lattice. The motivation for considering (1.1) is rooted in its natural interpretation in social choice: *the score of the median profile is the median of the scores of the profiles.*

Median algebras have been investigated by several authors (see [4, 9] for early references on median algebras and see [2, 10] for some surveys) who illustrated the deep interactions between median algebras, order theory and graph theory.

For instance, take an element a of a median algebra $\mathbf{A}$ and consider the relation $\leq_a$ defined on A by

$$x \leq_a y \quad\iff\quad \mathbf{m}(a, x, y) = x.$$

Endowed with this relation, $\mathbf{A}$ is a $\wedge$-semilattice order with bottom element a [13]: the associated operation $\wedge$ is defined by $x \wedge y = \mathbf{m}(a, x, y)$.

Semilattices constructed in this way are called *median semilattices*, and they coincide exactly with semilattices in which every principal ideal is a distributive lattice and in which any three elements have a join whenever each pair of them is bounded above. The operation $\mathbf{m}$ on $\mathbf{A}$ can be recovered from the median semilattice order $\leq_a$ using identity (1.3) where $\wedge$ and $\vee$ are defined with respect to $\leq_a$.

Note that if a median algebra $\mathbf{A}$ contains two elements 0 and 1 such that $\mathbf{m}(0, x, 1) = x$ for every $x \in A$, then $(A, \leq_0)$ is a distributive lattice order bounded by 0 and 1, and where $x \wedge y$ and $x \vee y$ are given by $\mathbf{m}(x, y, 0)$ and $\mathbf{m}(x, y, 1)$, respectively. Conversely, if $\mathbf{L} = \langle L, \vee, \wedge\rangle$ is a distributive lattice, then the term function defined by (1.3) is denoted by $\mathbf{m_L}$ and gives rise to a median algebra on L, called the *median algebra associated with* $\mathbf{L}$. It is noteworthy that equations satisfied by median algebras of the form $\langle L, \mathbf{m_L}\rangle$ are exactly those satisfied by median algebras. In particular, every median algebra satisfies the equation

$$\begin{aligned}\mathbf{m}(x, y, z) = \mathbf{m}\big(&\mathbf{m}\big(\mathbf{m}(x, y, z), x, t\big),\\ &\mathbf{m}\big(\mathbf{m}(x, y, z), z, t\big), \mathbf{m}\big(\mathbf{m}(x, y, z), y, t\big)\big). \qquad (1.4)\end{aligned}$$

Moreover, covering graphs (*i.e.*, undirected Hasse diagram) of median semilattices have been investigated and are, in a sense, equivalent to median graphs. Recall that a median graph is a (non necessarily finite) connected graph in which for any three vertices u, v, w there is exactly one vertex x that lies on a shortest path between u and v, on a shortest path between u and w and on a shortest path between v and w. In other words, x (the *median* of u, v and w) is the only vertex such that

$$\begin{aligned}d(u, v) &= d(u, x) + d(x, v),\\ d(u, w) &= d(u, x) + d(x, w),\\ d(v, w) &= d(v, x) + d(x, w).\end{aligned}$$

Every median semilattice whose intervals are finite has a median covering graph [1] and conversely, every median graph is the covering graph of a median semilattice [1, 13]. This connection is deeper: median semilattices can be characterized among the ordered sets whose bounded chains are finite and in which any two elements are bounded below as the ones

[1] LAMSADE - CNRS, Université Paris-Dauphine, Place du Maréchal de Lattre de Tassigny, 75775 Paris Cedex 16, France, and LORIA (CNRS – Inria Nancy Grand Est – Université de Lorraine), BP239, 54506 Vandoeuvre les Nancy, France, miguel.couceiro[at]inria.fr

[2] Mathematics Research Unit, FSTC, University of Luxembourg, 6, rue Coudenhove-Kalergi, L-1359 Luxembourg, Luxembourg, bruno.teheux[at]uni.lu

whose covering graph is median [3]. For further background see, e.g., [2].

Here we are particularly interested in solving equation (1.1) for median algebras $\mathbf{A}$ that are *conservative*, i.e., that satisfy

$$\mathbf{m}(x,y,z) \in \{x,y,z\}, \qquad x,y,z \in A. \tag{1.5}$$

This property essentially states that the aggregation procedure (in this case, a median) should pick one of its entries (e.g., the median candidate is one of the candidate).

Semilattices associated with conservative median algebras are called *conservative median semilattices.* It is not difficult to verify that a median algebra is conservative if and only if each of its subsets is a median subalgebra. Moreover, if $\mathbf{L}$ is a chain, then $\mathbf{m_L}$ satisfies (1.5); however the converse is not true. This fact was observed in §11 of [12], which presents the four element Boolean algebra as a counter-example.

The results of this paper are twofold. First, we present a description of conservative median algebras in terms of forbidden substructures (in complete analogy with BIRKHOFF's characterization of distributive lattices with M_5 and N_5 as forbidden substructures and KURATOWSKI's characterization of planar graphs in terms of forbidden minors), and that leads to a representation of conservative median algebras (with at least five elements) as chains. In fact, the only conservative median algebra that is not representable as a chain is the four element Boolean algebra.

Second, we characterize functions $f : \mathbf{B} \to \mathbf{C}$ that satisfy the equation

$$f(\mathbf{m}(x,y,z)) \;=\; \mathbf{m}(f(x), f(y), f(z)), \tag{1.6}$$

where $\mathbf{B}$ and $\mathbf{C}$ are finite products of (non necessarily finite) chains, as superposition of compositions of monotone maps with projection maps (Theorem 3.5). Particularized to aggregation functions $f : \mathbf{A}^n \to \mathbf{A}$, where $\mathbf{A}$ is a chain, we obtain an ARROW-like theorem: *f satisfies equation* (1.1) *if and only if it is dictatorial and monotone* (Corollary 3.6).

Throughout the paper we employ the following notation. For each positive integer n, we set $[n] = \{1, \ldots, n\}$. Algebras are denoted by bold roman capital letters $\mathbf{A}, \mathbf{B}, \mathbf{X}, \mathbf{Y} \ldots$ and their universes by italic roman capital letters $A, B, X, Y \ldots$. To simplify our presentation, we will keep the introduction of background to a minimum, and we will assume that the reader is familiar with the theory of lattices and ordered sets. We refer the reader to [6, 8] for further background. To improve the readability of the paper, we adopt the rather unusual convention that in any distributive lattice the empty set is a prime filter and a prime ideal. Proofs of the results presented in the third section are omitted because they rely on arguments involving a categorical duality that are beyond the scope of this paper.

2 Characterizations of conservative median algebras

Let $\mathbf{C}_0 = \langle C_0, \leq_0, c_0 \rangle$ and $\mathbf{C}_1 = \langle C_1, \leq_1, c_1 \rangle$ be chains with bottom elements c_0 and c_1. The $\perp$*-coalesced sum* $\mathbf{C}_0 \perp \mathbf{C}_1$ of $\mathbf{C}_0$ and $\mathbf{C}_1$ is the poset obtained by amalgamating c_0 and c_1 in the disjoint union of C_0 and C_1. Formally,

$$\mathbf{C}_0 \perp \mathbf{C}_1 = \langle C_0 \sqcup C_1 \,/\!\equiv,\ \leq \rangle,$$

where $\sqcup$ is the disjoint union, where $\equiv$ is the equivalence generated by $\{(c_0, c_1)\}$ and where $\leq$ is defined by

$$x/\!\equiv\ \leq\ y/\!\equiv \iff (x \in \{c_0, c_1\} \text{ or } x \leq_0 y \text{ or } x \leq_1 y).$$

Proposition 2.1. *The partially ordered sets $\mathbf{A}_1, \ldots, \mathbf{A}_4$ depicted in Fig. 1 are not conservative median semilattices.*

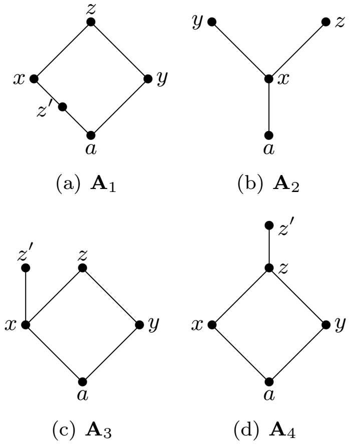


Figure 1. Examples of $\wedge$-semilattices that are not conservative.

Proof. The poset $\mathbf{A}_1$ is a bounded lattice (also denoted by N_5 in the literature on lattice theory, e.g., in [6, 8]) that is not distributive. In $\mathbf{A}_2$ the center is equal to the median of the other three elements. The poset $\mathbf{A}_3$ contains a copy of $\mathbf{A}_2$, and $\mathbf{A}_4$ is a distributive lattice that contains a copy of the dual of $\mathbf{A}_2$ and thus it is not conservative as a median algebra. □

The following Theorem provides descriptions of conservative semilattices with at least five elements, both in terms of forbidden substructures and in the form of representations by chains. Note that any semillatice with at most four elements is conservative, but the poset depicted in Fig. 1(b).

Theorem 2.2. *Let $\mathbf{A}$ be a median algebra with $|A| \geq 5$. The following conditions are equivalent.*

(1) *$\mathbf{A}$ is conservative.*
(2) *For every $a \in A$ the ordered set $\langle A, \leq_a \rangle$ does not contain a copy of the poset depicted in Fig. 1(b).*
(3) *There is an $a \in A$ and lower bounded chains $\mathbf{C}_0$ and $\mathbf{C}_1$ such that $\langle A, \leq_a \rangle$ is isomorphic to $\mathbf{C}_0 \perp \mathbf{C}_1$.*
(4) *For every $a \in A$, there are lower bounded chains $\mathbf{C}_0$ and $\mathbf{C}_1$ such that $\langle A, \leq_a \rangle$ is isomorphic to $\mathbf{C}_0 \perp \mathbf{C}_1$.*

Proof. (1) $\implies$ (2): Follows from Proposition 2.1.

(2) $\implies$ (1): Suppose that $\mathbf{A}$ is not conservative, that is, there are $a, b, c, d \in A$ such that $d := \mathbf{m}(a, b, c) \notin \{a, b, c\}$. Clearly, a, b and c must be pairwise distinct. By (1.2), a and b are $\leq_c$-incomparable, and $d <_c a$ and $d <_c b$. Moreover, $c <_c d$ and thus $\langle \{a, b, c, d\}, \leq_c \rangle$ is a copy of A_2 in $\langle A, \leq_c \rangle$.

(1) $\implies$ (4): Let $a \in A$. First, suppose that for every $x, y \in A \setminus \{a\}$ we have $\mathbf{m}(x, y, a) \neq a$. Since $\mathbf{A}$ is conservative,

for every $x, y \in A$, either $x \leq_a y$ or $y \leq_a x$. Thus $\leq_a$ is a chain with bottom element a, and we can choose $\mathbf{C}_1 = \langle A, \leq_a, a\rangle$ and $\mathbf{C}_2 = \langle \{a\}, \leq_a, a\rangle$.

Suppose now that there are $x, y \in A \setminus \{a\}$ such that $\mathbf{m}(x, y, a) = a$, that is, $x \wedge y = a$. We show that

$$z \neq a \implies \big(\mathbf{m}(x, z, a) \neq a \text{ or } \mathbf{m}(y, z, a) \neq a\big), \quad z \in A. \tag{2.1}$$

For the sake of a contradiction, suppose that $\mathbf{m}(x, z, a) = a$ and $\mathbf{m}(y, z, a) = a$ for some $z \neq a$. By equation (1.4), we have

$$\begin{aligned}\mathbf{m}(x, y, z) = \mathbf{m}\big(&\mathbf{m}\big(\mathbf{m}(x, y, z), x, a\big),\\ &\mathbf{m}\big(\mathbf{m}(x, y, z), z, a\big), \mathbf{m}\big(\mathbf{m}(x, y, z), y, a\big)\big).\end{aligned} \tag{2.2}$$

Assume that $\mathbf{m}(x, y, z) = x$. Then (2.2) is equivalent to

$$x = \mathbf{m}(x, \mathbf{m}(x, z, a), \mathbf{m}(x, y, a)) = a,$$

which yields the desired contradiction. By symmetry, we derive the same contradiction in the case $\mathbf{m}(x, y, z) \in \{y, z\}$.

We now prove that

$$z \neq a \implies \big(\mathbf{m}(x, z, a) = a \text{ or } \mathbf{m}(y, z, a) = a\big), \quad z \in A. \tag{2.3}$$

For the sake of a contradiction, suppose that $\mathbf{m}(x, z, a) \neq a$ and $\mathbf{m}(y, z, a) \neq a$ for some $z \neq a$. Since $\mathbf{m}(x, y, a) = a$ we have that $z \notin \{x, y\}$.

If $\mathbf{m}(x, z, a) = z$ and $\mathbf{m}(y, z, a) = y$, then $y \leq_a z \leq_a x$ which contradicts $x \wedge y = a$. Similarly, if $\mathbf{m}(x, z, a) = z$ and $\mathbf{m}(y, z, a) = z$, then $z \leq_a x$ and $z \leq_a y$ which also contradicts $x \wedge y = a$. The case $\mathbf{m}(x, z, a) = x$ and $\mathbf{m}(y, z, a) = z$ leads to similar contradictions.

Hence $\mathbf{m}(x, z, a) = x$ and $\mathbf{m}(y, z, a) = y$, and the $\leq_a$-median semilattice arising from the subalgebra $\mathbf{B} = \{a, x, y, z\}$ of $\mathbf{A}$ is the median semilattice associated with the four element Boolean algebra. Let $z' \in A \setminus \{a, x, y, z\}$. By (2.1) and symmetry we may assume that $\mathbf{m}(x, z', a) \in \{x, z'\}$. First, suppose that $\mathbf{m}(x, z', a) = z'$. Then $\langle \{a, x, y, z, z'\}, \leq_a\rangle$ is N_5 (Fig. 1(a)) which is not a median semilattice. Suppose then that $\mathbf{m}(x, z', a) = x$. In this case, the restriction of $\leq_a$ to $\{a, x, y, z, z'\}$ is depicted in Fig. 1(c) or 1(d), which contradicts Proposition 2.1, and the proof of (2.3) is thus complete.

Now, let $C_0 = \{z \in A \mid (x, z, a) \neq a\}$, $C_1 = \{z \in A \mid (y, z, a) \neq a\}$ and let $\mathbf{C}_0 = \langle C_0, \leq_a, a\rangle$ and $\mathbf{C}_1 = \langle C_1, \leq_a, a\rangle$. It follows from (2.1) and (2.3) that $\langle \mathbf{A}, \leq_a\rangle$ is isomorphic to $\mathbf{C}_0 \bot \mathbf{C}_1$.

(4) $\implies$ (3): Trivial.

(3) $\implies$ (1): Let $x, y, z \in \mathbf{C}_0 \bot \mathbf{C}_1$. If $x, y, z \in C_i$ for some $i \in \{0, 1\}$ then $\mathbf{m}(x, y, z) \in \{x, y, z\}$. Otherwise, if $x, y \in C_i$ and $z \notin C_i$, then $\mathbf{m}(x, y, z) \in \{x, y\}$. □

The equivalence between (3) and (1) gives rise to the following representation of conservative median algebras.

Theorem 2.3. *Let* $\mathbf{A}$ *be a median algebra with* $|A| \geq 5$. *Then* $\mathbf{A}$ *is conservative if and only if there is a totally ordered set* $\mathbf{C}$ *such that* $\mathbf{A}$ *is isomorphic to* $\langle \mathbf{C}, \mathbf{m}_{\mathbf{C}}\rangle$.

Proof. Sufficiency is trivial. For necessity, consider the universe of $\mathbf{C}_0 \bot \mathbf{C}_1$ in condition (3) endowed with $\leq$ defined by $x \leq y$ if $x \in C_1$ and $y \in C_0$ or $x, y \in C_0$ and $x \leq_0 y$ or $x, y \in C_1$ and $y \leq_1 x$. □

As stated in the next result, the totally ordered set $\mathbf{C}$ given in Theorem 2.3 is unique, up to (dual) isomorphism.

Theorem 2.4. *Let* $\mathbf{A}$ *be a median algebra. If* $\mathbf{C}$ *and* $\mathbf{C}'$ *are two chains such that* $\mathbf{A} \cong \langle \mathbf{C}, \mathbf{m}_{\mathbf{C}}\rangle$ *and* $\mathbf{A} \cong \langle \mathbf{C}', \mathbf{m}_{\mathbf{C}'}\rangle$, *then* $\mathbf{C}$ *is order isomorphic or dual order isomorphic to* $\mathbf{C}'$.

3 Homomorphisms between conservative median algebras

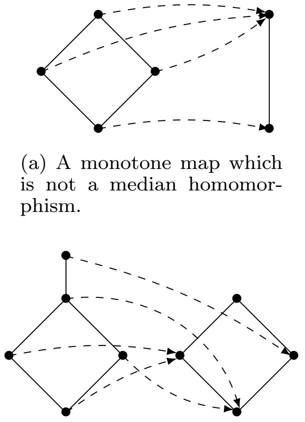
In view of Theorem 2.3 and Theorem 2.4, we introduce the following notation. Given a conservative median algebra $\mathbf{A}$ ($|A| \geq 5$), we denote a chain representation of $\mathbf{A}$ by $\mathbf{C}(\mathbf{A})$, that is, $\mathbf{C}(\mathbf{A})$ is a chain such that $\mathbf{A} \cong \langle \mathbf{C}(\mathbf{A}), \mathbf{m}_{\mathbf{C}(\mathbf{A})}\rangle$, and we denote the corresponding isomorphism by $f_{\mathbf{A}} : \mathbf{A} \to \langle \mathbf{C}(\mathbf{A}), \mathbf{m}_{\mathbf{C}(\mathbf{A})}\rangle$. If $f : \mathbf{A} \to \mathbf{B}$ is a map between two conservative median algebras with at least five elements, the map $f' : \mathbf{C}(\mathbf{A}) \to \mathbf{C}(\mathbf{B})$ defined as $f' = f_{\mathbf{B}} \circ f \circ f_{\mathbf{A}}^{-1}$ is said to be *induced by* f.

A function $f : \mathbf{A} \to \mathbf{B}$ between median algebras $\mathbf{A}$ and $\mathbf{B}$ is called a *median homomorphism* if it satisfies equation (1.6). We use the terminology introduced above to characterize median homomorphisms between conservative median algebras. Recall that a map between two posets is *monotone* if it is isotone or antitone.

Theorem 3.1. *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be two conservative median algebras with at least five elements. A map* $f : \mathbf{A} \to \mathbf{B}$ *is a median homomorphism if and only if the induced map* $f' : \mathbf{C}(\mathbf{A}) \to \mathbf{C}(\mathbf{B})$ *is monotone.*

Corollary 3.2. *Let* $\mathbf{C}$ *and* $\mathbf{C}'$ *be two chains. A map* $f : \mathbf{C} \to \mathbf{C}'$ *is a median homomorphism if and only if it is monotone.*

Remark 3.3. Note that Corollary 3.2 only holds for chains. Indeed, Fig. 2(a) gives an example of a monotone map that is not a median homomorphism, and Fig. 2(b) gives an example of median homomorphism that is not monotone.



(a) A monotone map which is not a median homomorphism.

(b) A median homomorphism which is not monotone.

Figure 2. Examples for Remark 3.3.

Since the class of conservative median algebras is clearly closed under homomorphic images, we obtain the following corollary.

Corollary 3.4. *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be two median algebras and* $f : \mathbf{A} \to \mathbf{B}$. *If* $\mathbf{A}$ *is conservative, and if* $|A|, |f(A)| \geq 5$, *then* f *is a median homomorphism if and only if* $f(\mathbf{A})$ *is a conservative median subalgebra of* $\mathbf{B}$ *and the induced map* $f' : \mathbf{C}(\mathbf{A}) \to \mathbf{C}(f(\mathbf{A}))$ *is monotone.*

We are actually able to lift the previous result to finite products of chains. If $f_i : A_i \to A'_i$ ($i \in [n]$) is a family of maps, let $(f_1, \ldots, f_n) : A_1 \times \cdots \times A_n \to A'_1 \times \cdots \times A'_n$ be defined by

$$(f_1, \ldots, f_n)(x_1, \ldots, x_n) := (f_1(x_1), \ldots, f_n(x_n)).$$

If $A = A_1 \times \cdots \times A_n$ and $i \in [n]$, then we denote the projection map from A onto A_i by π_i^A, or simply by π_i if there is no danger of ambiguity.

The following theorem characterizes median homomorphisms between finite products of chains.

Theorem 3.5. *Let* $\mathbf{A} = \mathbf{C}_1 \times \cdots \times \mathbf{C}_k$ *and* $\mathbf{B} = \mathbf{D}_1 \times \cdots \times \mathbf{D}_n$ *be two finite products of chains. Then* $f : \mathbf{A} \to \mathbf{B}$ *is a median homomorphism if and only if there exist* $\sigma : [n] \to [k]$ *and monotone maps* $f_i : \mathbf{C}_{\sigma(i)} \to \mathbf{D}_i$ *for* $i \in [k]$ *such that* $f = (f_{\sigma(i)}, \ldots, f_{\sigma(k)})$.

As an immediate consequence, it follows that aggregation functions compatible with median functions on ordinal scales are dictorial.

Corollary 3.6. *Let* $\mathbf{C}_1, \ldots, \mathbf{C}_n$ *and* $\mathbf{D}$ *be chains. A map* $f : \mathbf{C}_1 \times \cdots \times \mathbf{C}_n \to \mathbf{D}$ *is a median homomorphism if and only if there is a* $j \in [n]$ *and a monotone map* $g : \mathbf{C}_j \to \mathbf{D}$ *such that* $f = g \circ \pi_j$.

In the particular case of Boolean algebras (*i.e.*, powers of a two element chain), Theorem 3.5 can be restated as follows.

Corollary 3.7. *Assume that* $f : \mathbf{A} \to \mathbf{B}$ *is a map between two finite Boolean algebras* $\mathbf{A} \cong \mathbf{2}^n$ *and* $\mathbf{B} \cong \mathbf{2}^m$.

1. *The map* f *is a median homomorphism if and only if there are* $\sigma : [m] \to ([n] \cup \{\bot\})$ *and* $\epsilon : [m] \to \{\mathrm{id}, \neg\}$ *such that*

   $$f : (x_1, \ldots, x_n) \mapsto (\epsilon_1 x_{\sigma_1}, \ldots, \epsilon_m x_{\sigma_m}),$$

   where $x_\bot$ *is defined as the constant map* 0.

In particular,

2. *A map* $f\colon \mathbf{A} \to \mathbf{2}$ *is a median homomorphism if and only if it is a constant function, a projection map or the negation of a projection map.*
3. *A map* $f : \mathbf{A} \to \mathbf{A}$ *is a median isomorphism if and only if there is a permutation* σ *of* $[n]$ *and an element* ϵ *of* $\{\mathrm{id}, \neg\}^n$ *such that* $f(x_1, \ldots, x_n) = (\epsilon_1 x_{\sigma(1)}, \ldots, \epsilon_n x_{\sigma(n)})$ *for any* $(x_1, \ldots, x_n)$ *in* A.

4 Acknowledgment

This work was supported by the internal research project F1R-MTHPUL-12RDO2 of the University of Luxembourg.

References

[1] S. P. Avann. Metric ternary distributive semi-lattices. *Proceedings of the American Mathematical Society*, 12:407–414, 1961

[2] H. J. Bandelt and J. Hedlíková. Median algebras. *Discrete mathematics*, 45:1–30, 1983.

[3] H. J. Bandelt. Discrete ordered sets whose covering graphs are median. *Proceedings of the American Mathematical Society*, 91(1):6–8, 1984.

[4] G. Birkhoff and S. A. Kiss. A ternary operation in distributive lattices. *Bulletin of the American Mathematical Society*, 53:749–752, 1947.

[5] D. M. Clark and B. A. Davey. *Natural dualities for the working algebraist*, volume 57 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 1998.

[6] B. A. Davey and H. A. Priestley. *Introduction to lattices and order*. Cambridge University Press, New York, second edition, 2002.

[7] M. Grabisch, J.-L. Marichal, R. Mesiar, and E. Pap. *Aggregation functions*. Encyclopedia of Mathematics and its Applications, vol. 127. Cambridge University Press, Cambridge, 2009.

[8] G. Grätzer. *General lattice theory*. Birkhäuser Verlag, Basel, second edition, 1998. New appendices by the author with B. A. Davey, R. Freese, B. Ganter, M. Greferath, P. Jipsen, H. A. Priestley, H. Rose, E. T. Schmidt, S. E. Schmidt, F. Wehrung and R. Wille.

[9] A. A. Grau. Ternary Boolean algebra. *Bulletin of the American Mathematical Society*, (May 1944):567–572, 1947.

[10] J. R. Isbell. Median algebra. *Transactions of the American Mathematical Society*, 260(2):319–362, 1980.

[11] J. Nieminen. The ideal structure of simple ternary algebras. *Colloquium Mathematicum*, 40(1):23–29, 1978/79.

[12] M. Sholander. Trees, lattices, order, and betweenness. *Proceedings of the American Mathematical Society*, 3(3):369–381, 1952.

[13] M. Sholander. Medians, lattices, and trees. *Proceedings of the American Mathematical Society*, 5(5):808–812, 1954.

[14] H. Werner. A duality for weakly associative lattices. In *Finite algebra and multiple-valued logic (Szeged, 1979)*, volume 28 of *Colloquia Mathematica Societatis János Bolyai*, pages 781–808. North-Holland, Amsterdam, 1981.

A Metaheuristic Approach for Preference Learning in Multi-Criteria Ranking based on Reference Points

Jinyan Liu, Wassila Ouerdane, Vincent Mousseau [1]

Abstract. In this paper, we are interested in a family of multi-criteria ranking methods called Ranking with Multiple reference Points (RMP). This method is based on pairwise comparisons, but instead of directly comparing any pair of alternatives, it compares rather the alternatives to a set of predefined reference points. We actually focus on a Simplified RMP model (S-RMP) in which the preference parameters include the criteria weights and the set of reference points ordered by importance. Elicitation of the parameters (from the data provided by the decision makers) leads us to the preference learning algorithms that cannot only be applied on relatively small dataset. Therefore, we propose in this work a preference learning methodology for learning S-RMP models from a large set of pairwise comparisons of alternatives. The newly proposed algorithm is a combination of an Evolutionary Algorithm and a Linear Programming approach. Empirical results and analysis are also presented.

1 Introduction

Decision makers (DMs) often face decision situations in which different conflicting viewpoints are to be considered. When modeling a real world decision problem using Multiple Criteria Decision Aid (MCDA) theories, several problematics can be considered: choice, sorting or ranking problem [18]. In this paper, we are interested in multi-criteria ranking problem, where the aim is to establish a preference order (or ranking) on the set of alternatives.

The field of MCDA offers a selection of methods and operational tools i.e. aggregation models that explicitly account for the diversity of the viewpoints considered. Each method constructs first a model of DM's preferences and then exploits this model in order to work out a recommendation. During the aggregation phase, different parameters are needed, such as weights, marginal value functions, thresholds, etc. depending on the method. Such parameters allow to elaborate models taking into account the DMs preferences. Therefore, it is clear that such preferences play a key role in the construction of the recommendations. In fact, they are meaningful and acceptable only if the DMs values are taken into account.

The process by which the DMs values or the parameters of an aggregation model are captured is called preference elicitation or disaggregation [18]. Preference elicitation aims at helping the analyst to appropriately elicit the DM's preferences to be represented in a decision model. The process consists of a dialogue with the DM, where the aim is to infer the values of the parameters from the holistic information given by the DM. We note that this is not an easy task, especially when the information provided contains inconsistencies. Moreover, the DM has in general a limited knowledge on the aggregation models and can only express his preferences in a rather intuitive and ambiguous manner.

Generally speaking, there are two paradigms of preference elicitation approaches, namely direct and indirect paradigms [13]. In the direct aggregation paradigm, the parameter values are directly provided by the DM through an interactive communication with the analyst. The aggregation model is firstly constructed with these parameters and then applied to the alternative set to obtain the DM's preferences. Within such a paradigm, the DM should make enough effort to understand the meaning and the roles of these parameters and to associate appropriate values to them, which may be beyond his cognitive limitation. In the indirect disaggregation paradigm, the DM provides holistic preference information such as pairwise comparisons, assignment examples etc., from which a preference model is derived and then applied to contextual recommendation. In contrast with the direct elicitation, the parameter values are regressed from the DM [10].

Nowadays, we are facing with decision problems involving large datasets. It requires adapting and improving the algorithms and technics to construct acceptable recommendations. Some of related challenges have been addressed in the scientific community of Preference Learning, which focus on the computational complexity rather than the decision problem itself [9]. In the perspective of preference elicitation, the emergence of applications in MCDA with the intention of dealing with large datasets provokes our research interests. In this perspective, we intend to provide an efficient algorithms to infer the parameter values of an aggregation model called S-RMP such that the DM has provided as input a large set of pairwise comparisons.

The paper is organized as follows: Section 2 introduces the RMP method and its simplified version S-RMP. We present our metaheuristic approach for S-RMP disaggregation in Section 3. The numerical analysis and the benefits of the proposed approach are provided in Section 4. At the end, we conclude the paper.

2 Ranking with Multiple reference Points

Recently, a ranking method called Ranking with Multiple reference Points (RMP) has been proposed in [16, 17]. The idea is to construct the global preference relation between two alternatives on the basis of their relative comparisons with specified reference points. This paper is concerned with learning the parameters of this method. We give in the next sections an overview of such a method as well as its simplified version named S-RMP, introduced in [23, 22]. Further work has also been presented in [2].

[1] Laboratoire Génie Industriel, Ecole Centrale Paris, Grande Voie des Vignes, 92295 Châtenay-Malabry, e-mail: jinyan.liu@ecp.fr, wassila.ouerdane@ecp.fr, vincent.mousseau@ecp.fr

2.1 General RMP method

We consider a multiple criteria ranking problem with n alternatives in the set $\mathcal{A}$ indexed by $N = \{1, 2, ..., i, ..., n\}$ and m monotone criteria in the set $\mathcal{F}$, indexed by means of a set $M = \{1, 2, ..., j, ..., m\}$. The evaluations of alternatives on a criterion j take their value in the associated evaluation scale $\mathcal{X}_j$. The $\mathcal{X}$ denotes the evaluation space, $\mathcal{X} = \prod_{j \in M} \mathcal{X}_j$ i.e. the Cartesian product of evalution scales. Obviously, the evaluation of any alternative $a \in \mathcal{A}$ is a vector denoted by $a = (a_1, a_2, ..., a_m) \in \mathcal{X}$.

The RMP method is a three-step multi-criteria paradigm for ranking alternatives. It involves k reference points such that $P = \{1, 2, ..., h, ..., k\}$. The evaluation of each reference point $p^h, h \in P$ on a criterion j is denoted by $p^h_j \in \mathcal{X}_j$. To establish a global preference relation between two alternatives, a and b, the method specifies the following three steps:

1. Compare each alternative $a \in \mathcal{A}$ (respectively, $b \in \mathcal{A}$) to every reference points $p^h, h \in P$ on every criterion $j, j \in M$.
2. Aggregate the results of the step 1 considering the m criteria and deduce the preference relation between two different alternatives (a, b) which is depending on the reference point p^h, also called the relative preference with respect to the reference point p^h;
3. For each pair of alternatives (a, b), aggregate the k preference relations into global preference relation.

The first step establishes the preference relation between each alternative and each reference point on each criterion j. In the second, we only consider the criteria for which a (respectively, b) is at least as good as p^h. This set of criteria is denoted by $C(a, p^h)$ (respectively, $C(b, p^h)$) and defined as $C(a, p^h) \in \mathcal{P}(M)$ such that $C(a, p^h) = \{j \in M \mid a_j \geqslant p^h_j\}$.

We define then the *importance relation among criteria with respect to the reference point*, denoted by $\underline{\blacktriangleright}_{p^h}$, which means that a set of criteria is at least as important as another set of criteria as follows;

Definition 1. *(Importance relation among criteria w.r.t. reference point)*

The importance relation $\underline{\blacktriangleright}_{p^h}$ is defined on $M \times M$ such that:

1. $\forall M_1 \subset M, M_1 \neq \emptyset \Rightarrow M_1 \underline{\blacktriangleright}_{p^h} \emptyset, \forall h \in P$

2. $\forall M_1 \subset M \Rightarrow M \underline{\blacktriangleright}_{p^h} M_1, \forall h \in P$

3. $\forall M_1, M_2 \subset M, M_1 \subset M_2 \Rightarrow M_2 \underline{\blacktriangleright}_{p^h} M_1, \forall h \in P$

Thus, the (relative) preference relation $\succsim_{p^h}$ defined in Definition 2 expresses how a pair of alternatives compare with each other with respect to the reference point p^h.

Definition 2. *(Relative preference w.r.t. reference point)*

The relative preference with respect to the reference point $\succsim_{p^h}$ on $\mathcal{A} \times \mathcal{A}$ is defined by:

$$a \succsim_{p^h} b \Leftrightarrow C(a, p^h) \underline{\blacktriangleright}_{p^h} C(b, p^h)$$

In the third step, we define firstly the importance relation $\trianglerighteq$ as below.

Definition 3. *(Importance relation among reference points)*

The importance relation $\trianglerighteq$ is defined on $P \times P$ such that:

1. $\forall P_1 \subset P, P_1 \neq \emptyset \Rightarrow P_1 \trianglerighteq \emptyset$

2. $\forall P_1 \subset P \Rightarrow P \trianglerighteq P_1$

3. $\forall P_1, P_2 \subset P, P_1 \subset P_2 \Rightarrow P_2 \trianglerighteq P_1$

We deduce then the (global) preference relation $\succsim$ as described in the definition 4. It means that a is at least as good as b if the coalition of reference points $P(a, b)$ which affirms that a is at least as good as b is more important than the coalition of reference points $P(b, a)$ which affirms that b is at least as good as a.

Definition 4. *(Global preference relation)*

The global preference relation on $\mathcal{A} \times \mathcal{A}$ on the basis of the relative preferences is:

$$a \succsim b \Leftrightarrow P(a, b) \trianglerighteq P(b, a)$$

We note that there is no lack of generality to impose a dominance relation among reference points, as it was shows in [17]. There exists always an equivalent RMP model for any RMP model with k reference points such that:

$$\forall j \in M, \forall h, h' \in P, h > h' \Rightarrow p^h_j \geqslant p^{h'}_j \quad (1)$$

It means also that:

$$\forall a \in \mathcal{A}, \forall h, h' \in P, h > h' \Rightarrow C(a, p^h) \subseteq C(a, p^{h'}) \quad (2)$$

The two importance relations mentioned respectively in Definition 1 and Definition 3 can be rather general and built on the basis of different rules. Particularly, we present in the next section a simplified version of RMP method, namely S-RMP in which the importance relation $\underline{\blacktriangleright}_{p^h}$ is defined by a concordance rule for all reference points while a lexicography of dictatorial reference points is used as the importance relation $\trianglerighteq$.

2.2 Simplified RMP model

In this work, we focus on a simplified version of RMP, named S-RMP model, as considered in [23, 22].

Firstly, the importance relation $\underline{\blacktriangleright}_{p^h}$ on the criteria set is re-defined based on a concordance rule by using an additive decomposition as follow:

$$C(a, p^h) \underline{\blacktriangleright}_{p^h} C(b, p^h) \Leftrightarrow \sum_{j \in C(a, p^h)} \omega_j \geqslant \sum_{j \in C(b, p^h)} \omega_j \quad (3)$$

where ω_j represents the weight of the criterion j. Formally, the criteria weights are normalized to 1.

Secondly, as shown in [17], an important result derived from social choice theory indicates that the only importance relation, which aggregates the k relative preference relations (with respect to reference points) and leads to transitive relation on each possible set of alternatives, is obtained by a lexicographical order on the reference points. Therefore, a permutation σ on P is used lexicographically to determine the importance relations among the reference points so as to deduce globally the preference relations between alternatives. This is represented by (4).

$$\begin{aligned} a \succsim b \quad &\Leftrightarrow \quad \exists h^* \in P \text{ s.t. } a \succsim_{p^{\sigma(h^*)}} b \\ &\qquad \text{and } \forall h \in \{1, ..., h^* - 1\}, a \sim_{p^{\sigma(h)}} b \qquad (4) \\ a \sim b \quad &\Leftrightarrow \quad \forall h \in P, a \sim_{p^h} b \end{aligned}$$

In details, the first reference point is denoted by $p^{\sigma(1)}$, the second one by $p^{\sigma(2)}$ and so on. To compare a and b, we look at first the reference point $p^{\sigma(1)}$. If a is strictly preferred to b with respect to $p^{\sigma(1)}$, then we affirm globally that a is preferred to b without even considering the other reference points and similarly vice versa. However, if a and b are considered to be indifferent with respect to $p^{\sigma(1)}$, we shall proceed with the second reference point $p^{\sigma(2)}$, etc. Once the strict relative preference between a and b is confirmed, the global preference between a and b is affirmed. If we still cannot differentiate these two alternatives until the last reference point $p^{\sigma(k)}$ has been processed then they are considered globally as tied for conclusion.

The additive decomposition of importance relation for subsets of criteria shows that the preference relation between two alternatives (a, b) can be computed by:

$$\begin{aligned} a \succsim b \;&\Leftrightarrow\; \exists h^* \in P \text{ s.t.} \\ &\qquad \sum_{j \in C(a, p^{\sigma(h^*)})} \omega_j > \sum_{j \in C(b, p^{\sigma(h^*)})} \omega_j \\ &\qquad \text{and } \forall h \in \{1, \ldots, h^* - 1\}, \\ &\qquad \sum_{j \in C(a, p^{\sigma(h)})} \omega_j = \sum_{j \in C(b, p^{\sigma(h)})} \omega_j \\ a \sim b \;&\Leftrightarrow\; \forall h \in P, \\ &\qquad \sum_{j \in C(a, p^h)} \omega_j = \sum_{j \in C(b, p^h)} \omega_j \end{aligned} \tag{5}$$

2.3 Learning an S-RMP model

2.3.1 *Problem context*

We are interested in learning S-RMP models in an indirect way. To account for that, we assume that the information provided by the DM takes form of pairwise comparisons of alternatives. Learning such a model amounts to setting values for the criteria weights and the reference points which are detailed as follow:

- The normalized weights of criteria, $\omega_j, j \in M$.
- The number of reference points, denoted by k.
- The reference points, $p^h, h \in P$ where $p^h = (p^h_1, p^h_2, ..., p^h_m)$ is a vector in the evaluation space $p^h \in \mathcal{X}$.
- The lexicographic order of the reference points, defined as a permutation σ on the index set of the reference points P. For instance, $p^{\sigma(1)}$ is the first reference point to which alternatives are compared, $\exists h \in P, \sigma(1) = h$, etc.

To better understand the problem, we provide a brief review on the latest related researches.

2.3.2 *Literature review*

A first attempt to elicit indirectly the preference model from the holistic information given by the DM was presented in [14] for the ELECTRE TRI method. They considered the pessimistic assignment rule, and developed a non-linear optimization formulation to infer all the parameters from a set of assignment examples. A similar approach was presented in [11] for a simplified version of ELECTRE TRI named Majority Rule Sorting Model (MR-Sort). In [11], only a few number of categories was considered and there was no veto threshold characterizing the discordance effect. Later, in [4], an extension of the previous work for ELECTRE TRI method has been presented in the group decision problems. The parameters were inferred from assignment examples, provided by multiple decision makers, based on a Mixed Integer Programming (MIP).

For the S-RMP method, [23] was the first work that proposed an MIP to infer the parameters of this models from a given set of pairwise comparisons. They assumed the existence of an S-RMP model that is able to correctly restore all the pairwise comparisons given by the DM, and provided a linearized set of constraints with binary variables. However, the algorithm proposed in [23] suffers from a limitation in the case where the pairwise comparisons provided by the DM contains inconsistent information. It cannot find any solution.

To overcome this problem, [12] proposed a new algorithm dealing with inconsistencies in the elicitation process. The proposal is also based on an MIP with binary variables. However, the algorithm searches for an optimal solution that is compatible with as many as possible pairwise comparisons provided by the DM. Besides, it is able to identify the inconsistent information.

We remark a common issue for the two algorithms developed for S-RMP models that is the high computation time due to the introduction of the binary variables. Actually, when dealing with a very limited number of pairwise comparisons, the previous two algorithms are proved to be quite efficient. Nevertheless, the computation time increases exponentially when a large quantity of pairwise comparisons are provided.

Recently, [19] presented a well-adapted algorithm for learning the parameters of MR-Sort models from large datasets. It takes the advantages of a heuristic approach combined with a Linear Programming (LP). More details are provided in [20]. We highlight the general idea of this work for sorting problems which in our work are ranking ones instead.

Preference elicitation for RMP models was also encouraged in [2]. Besides, other works are generally interested in preference disaggregation. For interested readers, please refer to [8, 6, 7].

3 A metaheuristic approach

The different approaches presented in section 2.3.2 for learning S-RMP models have formulated the learning task as a Mixed Integer linear programming (MIP) optimization problem. However, a common observation is the considerable high computation time due to the introduction of binary variables [12, 22, 23].

In the case where the decision problem implies a large dataset, suchlike MIP consumes even more computation time and usually leads to the insolvability of the problem within the limited time. This type of problems is considered as *hard* optimization problems, which is defined as optimization problems that cannot be solved to optimality ([1]).

Thus, our proposal turns to a metaheuristic algorithm with the intention of dealing efficiently with large datasets. In contrast to the MIPs, the metaheuristic will approximate as accurately as possible the parameters of the S-RMP model. We remark that multiple S-RMP models may exist. Our objective is to infer a satisfactory S-RMP model that is compatible with most (if not all of them) of the input information. An interactive model calibration process in the form of supplementary constraints on the parameters should be invoked in real decision cases.

3.1 Overview of the algorithm

The proposed metaheuristic follows the general idea of Evolutionary Algorithms (EA) and makes use of a local optimization. It starts with an initialized population of N_{model} S-RMP models rather than a single solution. *Evaluation*, *Selection*, *Mutation* and *Substitution* operations are iterated to adjust the parameters of each individual in the operating population. The iteration will repeats until either at least one of the individuals in the population is able to restore all the input pairwise comparisons or after a certain number of times. Actually, due to the limited computation time in practice, we interrupt the algorithm before it reach the optimality more often. In our context, the limited number of iterations is set to 100. It is summarized in Algorithm 1

Algorithm 1: OVERVIEW OF THE ALGORITHM

```
Initialize a population of $N_{model}$ S-RMP models
Evaluate each individual in the population
repeat
    Select the best individuals from the population with a probability $\xi$
    Adjust the reference points for selected individuals and yield
    mutants with a probability $\mu$
    Adjust the criteria weights for mutated individuals
    Evaluate the newly adjusted individuals
    Substitute the population with a probability $\zeta$ under the constraint
    of the initial population size
until at least one of the stopping criteria is met;
```

Actually, on the one side, the reference points are adjusted while yielding mutants. On the other side, the criteria weights are adjusted by solving an LP. The rest of this section is structured as below: Section 3.2 begins with the initialization of the reference points and the criteria weights. Section 3.3 presents how to learn the reference points. The LP for learning the criteria weights is provided in Section 3.4.

3.2 Initialization

The first step of the algorithm consists in initializing a population of N_{model} S-RMP models (in our context, $N_{model} = 10$). In metaheuristic algorithms, it is important that the initial population spans adequately the solution space as shown in [5]. To do this, we provide a primitive method.

3.2.1 Reference points

First, we assume that the number of reference points k is fixed beforehand. From a practical perspective, the number of reference points in S-RMP models never exceeds 3, since such an S-RMP model has already a considerable capacity while restoring preference information as shown in [22].

Second, for a criterion j, k random numbers are generated from the uniform distribution in the evaluation scale $\mathcal{X}_j$. Depending on the preference direction of the criterion j, they are then ranked either in ascending order such that $p_j^1 \leqslant p_j^2 \leqslant ... \leqslant p_j^k$ or in descending order such that $p_j^k \leqslant p_j^{k-1} \leqslant ... \leqslant p_j^1$. Thus, the generated reference points guarantee that $p^h \succ p^{h-1} \succ ... \succ p^1$, where $p^h = (p_1^h, p_2^h, ..., p_j^h, ..., p_m^h)$

Finally, for k reference points, there are $k!$ possible lexicographic orders. We randomly choose one of them and it is fixed once initialized.

3.2.2 Criteria weights

Concerning the weights, we use the approach of *Butler* [3]. Firstly, $m - 1$ random integer numbers d_i where $i \in \{1, ..., m-1\}$ are generated uniformly between 0 and 100. Then, they are ranked such that $d_0 = 0 < d_1 < d_2 < ... < d_{m-1} < 100 = d_m$. The weight of a criterion j is defined as $w_j = (d_j - d_{j-1})/100$. It is accurate to 0.01 and $0 \leqslant w_j \leqslant 1$. This ensure that the weights sum up to 1 and are uniformly distributed.

3.3 Learning the reference points

3.3.1 Evolutionary algorithm

Our algorithm starts with an initial population of a certain number of random solutions, and evolves it while yielding a new generation of population at each iteration. Conventionally, the $(t+1)$-th generation of solutions, denoted by $\mathcal{G}_{t+1}$, is obtained from the n-th generation, denoted by $\mathcal{G}_t$, through a procedure composed by several well-defined operations as described in [21]. [15] also proposed a review about Evolutionary Algorithms used in the context of local search.

In Algorithm 1, a proportion of the best S-RMP models in the population are selected at the beginning of each iteration. Then, they are submitted to the mutation operation. Mutations are performed on each selected model and yield mutants with a probability. The criteria weights of the mutants are adjusted (Section 3.4). After being re-evaluated, the mutants are considered as the ”*children*”, while the initial models are considered as the ”*parents*”. Finally, the substitution amounts to selecting the best models among both the parents and the children based on their evaluation. Otherwise, it also allows us to introduce newly initialized individuals to the next generation of population if necessary. The details are provided in the following sections.

3.3.2 Evaluation operation

The evaluation of the ”best” individuals is done according to their *fitness* to the problem. In the context of learning an appropriate preference model from a set of information provided by the DM, several objective functions are applicable. In the case of learning S-RMP model, the fitness function corresponds to the *Ranking Accuracy* (RA) of the model, which is simply defined as the ratio of the number of pairwise comparisons restored correctly by the model to the total number of pairwise comparisons provided by the DM at the beginning of the process ((6)). Actually, this is the most straightforward measure of ranking performance.

$$RA = \frac{\text{Number of pairwise comparisons restored}}{\text{Total number of pairwise comparisons provided}} \tag{6}$$

3.3.3 Selection operation

The S-RMP model that gives the highest RA in the current population $\mathcal{G}_t$ will be selected randomly with replacement, with a probability ξ which increases with their fitness. Thereby, we define the probability $\xi(f_s)$ associated to a given S-RMP model f_s that consists of a set of parameters s by:

$$\xi(f_s) = \frac{RA(f_s) - RA_{min}}{RA_{max} - RA_{min}} \tag{7}$$

where

$$RA_{min} = \min\{RA(f_s) \mid \forall f_s \in \mathcal{G}_t\} \tag{8}$$

$$RA_{max} = \max\{RA(f_s) \mid \forall f_s \in \mathcal{G}_t\} \tag{9}$$

3.3.4 Mutation operation

Mutation operation, which amounts to adjust the reference points, is only applied to the selected models. For each selected S-RMP model, the operation is defined in the Algorithm 2. The mutations of the reference points proceed one by one based on their lexicographic order σ and then on each criterion from $j = 1$ to m.

Algorithm 2: MUTATION OPERATION

```
Create an empty list for keeping ignorable pairwise comparisons
for each reference point $p^{\sigma(h)}$ do
    for each criterion $j$ do
        Generate a random variation $\pm\theta_j^{\sigma(h)}$
        Count possible impacts caused by this change
        Apply the change to the current model with a probability $\mu$
    Append newly ignorable pairwise comparisons to the list
Yield a mutated model
```

Firstly, both the sign and the value of the variation, denoted by $\pm\theta_j^{\sigma(h)}$, are uniformly randomized. However, it should be bounded to prevent re-degrading the quality of the models after some iterations and the boundary should depend on the current number of iterations N_{it}. For example, we take herein:

$$\theta_j^{\sigma(h)} \leq \left\lceil \frac{50}{\sqrt{N_{it}}} \right\rceil \tag{10}$$

As $\pm\theta_j^{\sigma(h)}$ will be later applied to the current model with a probability μ, we denote the changed evaluation of the reference point $p^{\sigma(h)}$ on the criterion j by $p'^{\sigma(h)}_j$ where

$$p'^{\sigma(h)}_j = p_j^{\sigma(h)} \pm \theta_j^{\sigma(h)} \tag{11}$$

Then, we should define the probability μ that indicates if the variation $\pm\theta_j^{\sigma(h)}$ should be applied. To do so, we identify at first the impacts provoked by $\pm\theta_j^{\sigma(h)}$ on the judgment of preference between alternatives, which eventually improve (or worsen) the RA, through the calculations below.

For any pairwise comparison of alternatives $a \succsim b$ provided by the DM, we define the *slack* quantity as follows:

$$s_{(a,b)}^{\sigma(h)} = \sum_{j=1}^{m} \left(\delta_{a,j}^{\sigma(h)} - \delta_{b,j}^{\sigma(h)} \right) \cdot \omega_j \tag{12}$$

where ω_j denotes the weight of the criterion j. $\forall (a,b) \in \mathcal{BC}$, $\forall h \in P$ and the lexicographic order σ, we compute $\delta_{a,j}^{\sigma(h)}$ and $\delta_{b,j}^{\sigma(h)}$ as follows:

$$\delta_{a,j}^{\sigma(h)} = \begin{cases} 1 & \text{if } a_j \geqslant p_j^{\sigma(h)} \\ 0 & \text{if } a_j < p_j^{\sigma(h)} \end{cases} \quad , \quad \delta_{b,j}^{\sigma(h)} = \begin{cases} 1 & \text{if } b_j \geqslant p_j^{\sigma(h)} \\ 0 & \text{if } b_j < p_j^{\sigma(h)} \end{cases} \tag{13}$$

In fact, the (global) judgement of preference between (a, b) is based on the relative preference between (a, b) with respect to the reference point $p^{\sigma(h)}$, which is deduced by

$$a \succsim_{\sigma(h)} b \Longleftrightarrow s_{(a,b)}^{\sigma(h)} \geq 0 \tag{14}$$

If $\delta_{a,j}^{\sigma(h)}$ equals to 1, it means that the criterion j is contributing to the statement $a \succsim_{\sigma(h)} b$. The criterion j is namely a contributing criterion for a in this case. Similarly, if $\delta_{b,j}^{\sigma(h)}$ equals to 1, it means that the criterion j is weakening the statement $a \succsim_{\sigma(h)} b$. The criterion j is then namely a weakening criterion for b. Particularly, when $\delta_{a,j}^{\sigma(h)} - \delta_{b,j}^{\sigma(h)} = 0$, the criterion j is neither contributing nor weakening. We said that it is neutralized.

The variation $\pm\theta_j^{\sigma(h)}$ affects the calculation of $s_{(a,b)}^{\sigma(h)}$ through $(\delta_{a,j}^{\sigma(h)}, \delta_{b,j}^{\sigma(h)})$. All the possible impacts provoked by $\pm\theta_j^{\sigma(h)}$ can be summarized (as shown in Table 1) by the value change of $(\delta_{a,j}^{\sigma(h)}, \delta_{b,j}^{\sigma(h)})$, where the changed value can be denoted and calculated by:

$$\delta'^{\sigma(h)}_{a,j} = \begin{cases} 1 & \text{if } a_j \geqslant p'^{\sigma(h)}_j \\ 0 & \text{if } a_j < p'^{\sigma(h)}_j \end{cases} \tag{15}$$

$$\delta'^{\sigma(h)}_{b,j} = \begin{cases} 1 & \text{if } b_j \geqslant p'^{\sigma(h)}_j \\ 0 & \text{if } b_j < p'^{\sigma(h)}_j \end{cases} \tag{16}$$

Formally, the positive and negative impacts are respectively defined as follows:

$$I_{pos}\left(\pm\theta_j^{\sigma(h)}\right) = \left\{ (a,b) \in \mathcal{BC} \;\middle|\; (\delta'^{\sigma(h)}_{a,j} - \delta'^{\sigma(h)}_{b,j}) - (\delta_{a,j}^{\sigma(h)} - \delta_{b,j}^{\sigma(h)}) > 0 \right\} \tag{17}$$

$$I_{neg}\left(\pm\theta_j^{\sigma(h)}\right) = \left\{ (a,b) \in \mathcal{BC} \;\middle|\; (\delta'^{\sigma(h)}_{a,j} - \delta'^{\sigma(h)}_{b,j}) - (\delta_{a,j}^{\sigma(h)} - \delta_{b,j}^{\sigma(h)}) < 0 \right\} \tag{18}$$

Table 1: Impacts provoked by $\pm\theta_j^{\sigma(h)}$ on $\left(\delta_{a,j}^{\sigma(h)}, \delta_{b,j}^{\sigma(h)}\right)$

	Before	After	Impact	Description
I_1	(0,1)	(0,0)	positive	weakening criterion neutralized
I_2	(0,1)	(1,1)	positive	weakening criterion neutralized
I_3	(0,0)	(1,0)	positive	criterion become contributing
I_4	(1,1)	(1,0)	positive	criterion become contributing
I_5	(1,0)	(0,0)	negative	contributing criterion neutralized
I_6	(1,0)	(1,1)	negative	contributing criterion neutralized
I_7	(0,0)	(0,1)	negative	criterion become weakening
I_8	(1,1)	(0,1)	negative	criterion become weakening
I_9	(0,0)	(1,1)	neutral	no impact on the slack
I_{10}	(1,1)	(0,0)	neutral	no impact on the slack
I_{11}	(0,0)	(0,0)	neutral	no impact on the slack
I_{12}	(1,1)	(1,1)	neutral	no impact on the slack
I_{13}	(1,0)	(1,0)	neutral	no impact on the slack
I_{14}	(0,1)	(0,1)	neutral	no impact on the slack
I_{15}	(0,1)	(1,0)	n.c.	impossible
I_{16}	(1,0)	(0,1)	n.c.	impossible

For instance, we represent, in Figure 1, a couple of examples to illustrate the adjustment of reference points. We suppose that

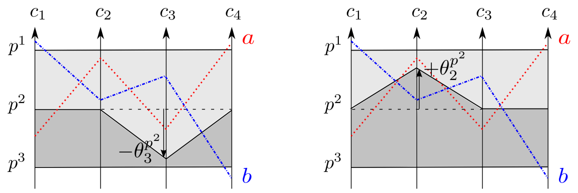
- The criteria (c_1, c_2, c_3 and c_4) are equal weighted, i.e. $\forall j \in \{1, 2, 3, 4\}, \omega_j = 0.25$.
- The reference points are uniformly initialized and used in sequence, i.e. first p^1, then p^2 and p^3.
- The DM provided $a \succsim b$ as input.

With the initialized reference points, we derive at first that $a \sim_{p^1} b$, $b \succsim_{p^2} a$ and $a \succsim_{p^3} b$. Since the reference points are used sequentially, we derive then $b \succsim a$, which is inconsistent with the statement of the DM. Thus, the reference points should be adjusted by the algorithm.

In Figure 1(a), the criterion $j = 3$, which was originally weakening the statement $a \succsim b$ (as $\delta^{p^2}_{a,3} = 0$), is neutralized by applying the variation $-\theta^{p^2}_3$ to the reference point p^2 on the criterion $j = 3$, because $\delta'^{p^2}_{a,3} = \delta'^{p^2}_{b,3} = 1$ and $\delta'^{p^2}_{a,3} - \delta'^{p^2}_{b,3} = 0$. It is considered to be a (potential) positive impact, for the fact that, as shown in this example, we newly derive that $a \sim_{p^2} b$ and finally $a \succsim b$, which is consistent with the DM's statement. In this sense, the reference point p^2 is adjusted.

In Figure 1(b), the criterion $j = 2$, which was originally neither weakening nor contributing to statement $a \succsim b$ (as $\delta^{p^2}_{a,2} - \delta^{p^2}_{b,2} = 0$), becomes contributing by applying the variation $+\theta^{p^2}_2$ to the reference point p^2 on the criterion $j = 2$, because $\delta'^{p^2}_{a,2} = 1$ and $\delta'^{p^2}_{b,2} = 0$. It is also considered to be a (potential) positive impact, for the similar fact that we can again finally derive that $a \succsim b$ in this example with the adjusted reference point p^2 even though it was adjusted in a different way.

Figure 1: Adjustment of reference points



(a) Impact of $-\theta^{p^2}_3$ (i.e. I_2)

(b) Impact of $+\theta^{p^2}_2$ (i.e. I_4)

So far, we identify four positive possibilities as well as four negative possibilities and sum up accordingly the quantity of the positive (respectively, negative) impacts by

$$I_{pos} = \left| I_{pos}\left(\pm\theta^{\sigma(h)}_j\right)\right| = |I_1| + |I_2| + |I_3| + |I_4| \tag{19}$$

$$I_{neg} = \left| I_{neg}\left(\pm\theta^{\sigma(h)}_j\right)\right| = |I_5| + |I_6| + |I_7| + |I_8| \tag{20}$$

However, we need to make some empirical remarks such that:

1. Usually, the variation $\pm\theta^{\sigma(h)}_j$ provokes both the positive and negative impacts on different pairs of alternatives at the same time. Having more positive impacts means that we have more chance to improve the S-RMP model quality, but not necessarily.
2. Sometimes, we provoke as many negative impacts as the positive impacts. In such a case, we are more interested in the case where a large quantity of impacts provoked, as it allows us to introduce more diversity to the population.
3. When there are less positive impacts than the negative ones, or even no positive impacts provoked, we consider that it is still possible to improve the model potentially.

Based on these observations, we define accordingly the probability of applying the variation $\pm\theta^{\sigma(h)}_j$ to the current S-RMP model, denoted by $\mu(\pm\theta^{\sigma(h)}_j)$ as it is associated to the variation $\pm\theta^{\sigma(h)}_j$, as below :

$$\mu\left(\pm\theta^{\sigma(h)}_j\right) = \begin{cases} \frac{I_{pos}}{I_{pos}+I_{neg}} & \text{if } I_{pos} \neq 0 \text{ and } I_{pos} > I_{neg} \\ 1 - \frac{1}{I_{pos}} & \text{if } I_{pos} \neq 0 \text{ and } I_{pos} = I_{neg} \\ 0.5 * Gaussian & \text{if } I_{pos} = 0 \text{ or } I_{pos} < I_{neg} \end{cases} \tag{21}$$

Otherwise, as we treat the reference points one by one according to the lexicographic order, once (14) is affirmed, (a, b) will be appended to the list for keeping ignorable pairwise comparisons. Actually, (a, b) is considered to be ignorable, because the adjustment for the rest of the reference points will not affect the global judgement on the preference between (a, b) and the impacts provoked by $\pm\theta^{\sigma(h+1)}_j$ on $(\delta^{\sigma(h+1)}_{a,j}, \delta^{\sigma(h+1)}_{b,j})$ (and so on, if exists) can be then ignored. In other words, once the comparison between any two alternatives (a, b) is restored by the previous reference point, no matter how they compare to the rest of the reference points, it will not be changed.

3.3.5 *Substitution operation*

As presented in Section 3.3.1, the mutants with the adjusted criteria weights (Section 3.4 for details) will be re-evaluated and then appended to the n-th generation of population $\mathcal{G}_t$. The population that contains both the "*children*" and the "*parents*" is denoted by $\mathcal{G}'_t$. Substitution operation decides who will survive in the $(t+1)$-th generation of population.

Firstly, it is about selecting the best individuals from the n-th generation of population $\mathcal{G}'_t$. Therefore, the principle is the same as the selection operation presented in Section 3.3.3. We define the probability ζ associated to an S-RMP model f_s of being submitted to the $(t + 1)$-th generation by:

$$\zeta(f_s) = \frac{RA(f_s) - RA'_{min}}{RA'_{max} - RA'_{min}} \tag{22}$$

where,

$$RA'_{min} = \min\left\{RA(f_s) \mid \forall f_s \in \mathcal{G}'_t\right\} \tag{23}$$

$$RA'_{max} = \max\left\{RA(f_s) \mid \forall f_s \in \mathcal{G}'_t\right\} \tag{24}$$

Secondly, if there is not enough number of individuals that are good enough to be submitted, some newly initialized S-RMP models will be appended to the $(t + 1)$-th generation under the constraint of the initial size of the operating population.

After having defined the substitution operation, the evolutionary part of the algorithm for learning the reference points is completed. Now, we discuss how to adjust the criteria weights.

3.4 Learning the criteria weights

Assuming that the reference points are given, we consider the adjustment of criteria weights as a linear optimization problem.

We are interested in setting good values for the criteria weights ω_j, $j \in M$ to restore as many as possible pairwise comparisons provided by the DM. However, to do that we usually need to introduce binary variables into the program as it is described in [22]. In the case of a large dataset, using binary variables will considerably increase the computation time ([23, 12]). Therefore, we propose in this paper an alternative formulation without binary variables that has been proved to be quite efficient (Section 4 for the numerical test results).

We note at first that ω_j, $j \in M$ are the unknown variables to be adjusted. Considering then the *slack* defined by (12) and (13), we remark that it is actually an indicator that shows if the (relative) preference between two alternatives can be restored by the current S-RMP model. Hopefully, for any pairwise comparison (a, b) provided by the DM, there should be at least one of the slack variables $s_{(a,b)}^{\sigma(h)}$ such that

$$\exists h^* \in P, s_{(a,b)}^{\sigma(h^*)} > 0, s_{(a,b)}^{\sigma(1)} = s_{(a,b)}^{\sigma(2)} = ... = s_{(a,b)}^{\sigma(h^*-1)} = 0 \quad (25)$$

where σ is a specified lexicographic order of reference points.

However, we cannot translate (25) as part of the linear constraints of the problem, because it is not always true. It means that it is not always possible to find such an S-RMP model with the given reference points and the adjusted criteria weights that restores exactly all the provided pairwise comparisons.

In fact, our objective is, by adjusting the criteria weights, to make the model compatible with as many as possible pairwise comparisons provided by the DM. In this case, (25) should be integrated as part of the objective function but not as linear constraints.

Hence, to account for this, we define then another two auxiliary variables $s_{(a,b)}^{\sigma(h)+}$ and $s_{(a,b)}^{\sigma(h)-}$ for $\forall h \in P, \forall (a, b) \in \mathcal{BC}$ by:

$$s_{(a,b)}^{\sigma(h)} - s_{(a,b)}^{\sigma(h)+} + s_{(a,b)}^{\sigma(h)-} \geq 0 \quad (26)$$

where both of them are positive by definition. Actually, $s_{(a,b)}^{\sigma(h)+}$ represents the positive terms in the definition of $s_{(a,b)}^{h}$ that contributes to the statement. Respectively, $s_{(a,b)}^{\sigma(h)-}$ represents the negative terms in the definition of $s_{(a,b)}^{h}$ that weaken the statement.

Moreover, in order to maximize the number of pairwise comparisons correctly restored by the model, the objective function is defined as below:

$$\max \sum_{(a,b)\in\mathcal{BC}} \sum_{h=1}^{k} \underline{\omega}^{\sigma(h)} \cdot \left(s_{(a,b)}^{\sigma(h)+} - \alpha \cdot s_{(a,b)}^{\sigma(h)-} \right) \quad (27)$$

where we make use of two weighting system α and $\underline{\omega}^{\sigma(h)}$.

On the one hand, $\underline{\omega}^{\sigma(h)}$ for $\forall h \in P$ weights the reference points. We remark that, by carefully choosing their values, it is approximately equivalent to using the reference points in their lexicographic order. Actually, if $k \geqslant 2$, whatever the value of $\underline{\omega}^{\sigma(k)}$,

$$\underline{\omega}^{\sigma(k-h)} = \frac{1}{\epsilon} \cdot \sum_{i=k-h+1}^{k} \underline{\omega}^{\sigma(i)} \quad (28)$$

that depends on the small value for ϵ. ϵ represent the accuracy of the criteria weights that we impose in the solution. For example, if $\epsilon = 10^{-3}$ and we take $\underline{\omega}^{\sigma(3)} = 1$ for an S-RMP model with 3 reference points, then we can deduce that $\underline{\omega}^{\sigma(2)} = 10^3$ and $\underline{\omega}^{\sigma(1)} = 10^6 + 10^3$.

On the other hand, α balances the compensatory behavior between pairwise comparisons. We note that, to avoid using binary variables, such an objective function cannot garantee that the maximal number of pairwise comparisons can be correctly restored. However, by setting a big enough value for α (for example, $\alpha = 10^3$), it is able to reduce the compensatory effects efficiently.

We summarize the linear program as below:

$$\begin{aligned}
&\max \sum_{(a,b)\in\mathcal{BC}} \sum_{h=1}^{k} \underline{\omega}^{\sigma(h)} \cdot \left(s_{(a,b)}^{\sigma(h)+} - \alpha \cdot s_{(a,b)}^{\sigma(h)-} \right) \\
&\text{s.t.} \sum_{j=1}^{m} \omega_j = 1 \\
&\quad \forall (a,b) \in \mathcal{BC}, \forall h \in P, \\
&\qquad s_{(a,b)}^{\sigma(h)} = \sum_{j=1}^{m} \left(\delta_{a,j}^{\sigma(h)} - \delta_{b,j}^{\sigma(h)} \right) \cdot \omega_j \\
&\qquad s_{(a,b)}^{\sigma(h)} - s_{(a,b)}^{\sigma(h)+} + s_{(a,b)}^{\sigma(h)-} \geq 0 \\
&\qquad s_{(a,b)}^{\sigma(h)+} \geq 0, s_{(a,b)}^{\sigma(h)-} \geq 0
\end{aligned}$$

4 Numerical analysis

In this section, the proposed algorithm is further investigated numerically based on a large set of artificially generated data. Statistical techniques were used to demonstrate the advantages of the metaheuristic comparing with the conventional disaggregation methods for outranking models based on LP.

We are firstly concerned with the quality of the solution. For a suchlike evolutionary approach, it is represented by the "best" individual in the operating population. Secondly, there is usually a compromise between the optimality of the solution and the number of iterations it takes to reach a satisfactory result as we mentioned in Section 3.1. By simulating the different decision circumstances with a large quantity of randomly generated data, we investigate the improvement curves produced in each circumstance to better understand the behavior of the algorithm. Otherwise, we examine also the runtime characteristics of the proposed metaheuristic, since one of the interest of this work is to overcome the insolvability of the MIP-based disaggregation methods in dealing with large datasets (Section 2.3.2).

The different decision circumstances that we studied are discussed in the next section.

4.1 Experiments

In the experiments, we consider 1000 alternatives and a varying numbers of criteria (4, 6, 8 or 10). We generate randomly an initial S-RMP model to test the algorithm. It is denoted by f_β, as it is defined by a set of parameters β. It simulates the preference model of a fictitious DM. Then, 500 reference pairwise comparisons among the alternatives are derived from the initial model f_β based on their randomly generated evaluations on the criteria. We consider also different numbers of reference points in the initial model f_β (Ini. NRP, for Initial Number of Reference Points) to simulate different complexity levels of the DM's preference system.

The experiments are divided into two groups according to the different preset numbers of reference points in the inferred model (Inf. NRP): In Group A, we infer S-RMP models with only 1 reference point whatever the initial number of reference points in β. While, in Group B, we infer S-RMP models with exactly their initial number of reference points, so as to compare these two groups of experiment and better understand the importance of carefully setting a corresponding number of reference points in S-RMP disaggregation. The different cases that we considered are distinguished by:

- Number of criteria = 4, 6, 8 or 10
- Ini. NRP = 1,2 or 3
- Inf. NRP = 1 or the initial number

The experiments are then numbered by a trio-index as shown in Table 2 and Table 3. For instance, ”4.1.A” means that, we consider 4 criteria and the initial model is generated with 1 reference point and it is in the group ”A”. For each experiment, all the provided results are based on the average of 100 repeated trials under the same testing conditions [2].

4.2 Empirical results

4.2.1 *Quality of the solution*

The inferred model is denoted by $f_{\hat{\beta}^*}$ with some parameters $\hat{\beta}^*$. As shown in Table 2 and Table 3, the ”closeness” between the initial model f_β and the inferred model $f_{\hat{\beta}^*}$ is measured by the RA. Since we generated the input information without introducing any inconsistency, the measured value of RA is always expected to be as close as possible to 1.00. The ”Starting RA” represents the quality of the ”best” individual in the initialized population. The ”Final RA” represents the quality of the final inferred model $f_{\hat{\beta}^*}$, which is the ”best” individual in the population when the algorithm terminated. However, we remind that multiple non-identical individuals may give the same value of RA, as the solution is not unique (as presented in Section 2.3.1).

In the group A of experiments, we generated the initial models with 1, 2 or 3 reference points and inferred S-RMP models involving only 1 reference point (Table 2). We notice that the best final RA, 0.997, is observed in the experiment 4.1.A, while the worst is observed in the experiment 4.3.A. It shows, in general, the capability of the proposed metaheuristic while inferring an S-RMP model as simply as possible.

However, exceptions are observed in the experiment 10.3.A. The final RA is instead even greater than 10.1.A on average. The randomized starting models are also better than in the other two cases (10.1.A and 10.2.A). This will be further discussed (Section 4.3).

Table 2: Group A of experiments

Exp.	Num. Cri.	Ini. NRP	Inf. NRP	Starting RA	Final RA
4.1.A	4	1	1	0.807	0.997
4.2.A	4	2	1	0.746	0.904
4.3.A	4	3	1	0.735	0.880
6.1.A	6	1	1	0.813	0.987
6.2.A	6	2	1	0.761	0.934
6.3.A	6	3	1	0.741	0.912
8.1.A	8	1	1	0.810	0.969
8.2.A	8	2	1	0.760	0.940
8.3.A	8	3	1	0.746	0.925
10.1.A	10	1	1	0.805	0.960
10.2.A	10	2	1	0.761	0.931
10.3.A	10	3	1	0.810	0.969

In the group B of experiments, we assume that we can correctly set the number of reference points in the S-RMP models to be inferred (Table 3). At this time, the worst final RA is observed in the experiment 10.3.B. It is reasonable, since it is the most complicated case that has been considered in our analysis. However, 0.950, which means 95% of the 500 input pairwise comparisons could be restored correctly in the inferred S-RMP model, is already satisfactory enough for the disaggregation.

Table 3: Group B of experiments

Exp.	Num. Cri.	Ini. NRP	Inf. NRP	Starting RA	Final RA
4.1.B	4	1	1	0.807	0.997
4.2.B	4	2	2	0.810	0.983
4.3.B	4	3	3	0.819	0.973
6.1.B	6	1	1	0.813	0.987
6.2.B	6	2	2	0.798	0.966
6.3.B	6	3	3	0.796	0.968
8.1.B	8	1	1	0.810	0.969
8.2.B	8	2	2	0.780	0.961
8.3.B	8	3	3	0.784	0.961
10.1.B	10	1	1	0.805	0.964
10.2.B	10	2	2	0.777	0.952
10.3.B	10	3	3	0.775	0.950

By comparing with the group A, we observe that the starting RAs as well as the final RAs are significantly enhanced in the group B of experiments by presetting correctly the number of reference points, especially in the experiments where the initial models are generated with more than one reference points (except for 10.3.A and 10.3.B).

To better understand the behavior of the proposed algorithm, the improvement curves are drawn and investigated below.

4.2.2 *Behavior of the algorithm*

The improvement curves show not only the starting and the final point, but the evolution of RA in function of the number of iterations during the whole process. As shown in Figure 2 (for Group A) and Figure 3 (for Group B), the RA is improved and converges progressively. Without any surprise, the exceptional phenomenon that we pointed out in Section 4.2.1 for 10.3.A is also observed in Figure 2(d). The dotted curve that represents the case involving 3 reference points is instead placed above the curve that represents the case involving 1 reference point (Section 4.3 for further discussions).

For a fixed number of criteria, by comparing each of the subfigures in Group A with the ones in Group B (for example, Figure 2(a) with Figure 3(a)), we can firstly visualize the intensively reduced level of RA due to the improperly set number of reference points. Moreover, we notice that the reduced level is more significant when the model involves fewer criteria.

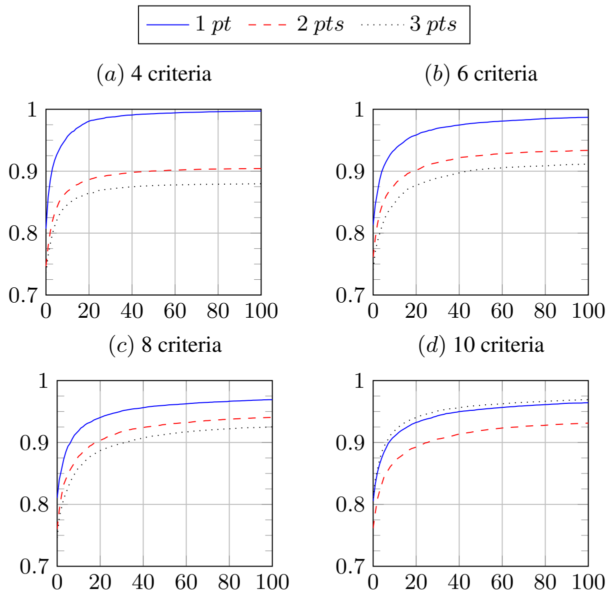
Otherwise, we observe that, in Figure 3, the dashed line for ”2p” and the dotted line for ”3p” almost coincide with each other. It means that, comparing with the single reference point cases, the improving RA is reduced quasi-equally in the multiple reference points cases regardless of the number of reference points involved.

Actually, the improvement curves can also be grouped by number of reference points (as shown in Figure 4 and Figure 5) instead of by criteria number. In Figure 4(c), we inferred S-RMP models that involve only 1 reference point from the information derived from the initial models that involve 3 reference points. We notice that, not similar to the other figures, the curves (in Figure 4(c)) that represent the cases where more criteria are involved are above the others that represent the cases where fewer criteria are involved. Related discussions are provided in Section 4.3.

Moreover, the improvement curves can be divided into two stages. The first stage is the rising section, while the second stage is the flat section. The flat section shows the final RA that we can reach,

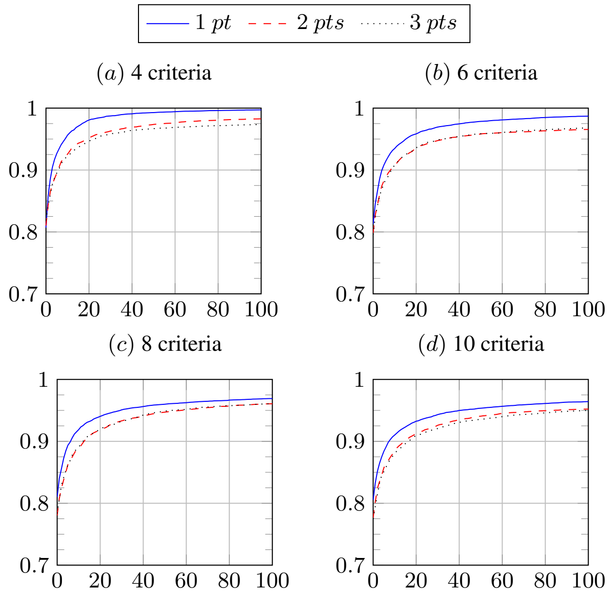
[2] Intel Core i3-2120 3.30 GHz CPU, 4 GB RAM, Ubuntu 14.04 LTS, Eclipse IDE Kepler SR2, Cplex 12.6

Figure 2: Group A of experiments (num. of criteria)



(a) 4 criteria (b) 6 criteria (c) 8 criteria (d) 10 criteria

Figure 3: Group B of experiments (num. of criteria)



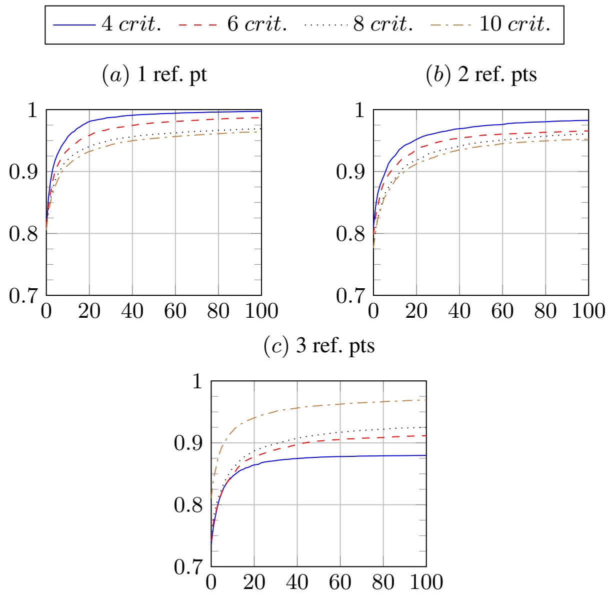
(a) 4 criteria (b) 6 criteria (c) 8 criteria (d) 10 criteria

while the rising section also reveals some important advantages of the proposed metaheuristic.

We measure the first stage of the improvement curves by the rapidity of convergence. It is defined as the necessary number of iterations needed to reach a quasi-satisfactory RA. To be precise, we define the quasi-satisfactory RA as the 97% of the final RA. It shows the efficiency of the metaheuristic algorithm. The results are presented in Table 4.

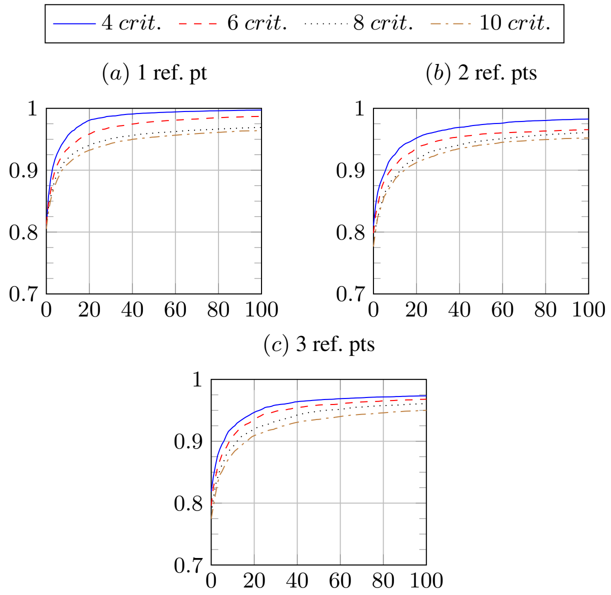
We can observe that the metaheuristic approach is generally quite efficient. Table 4 shows that the inferred models can reach a quasi-satisfactory state within around the first 30 iterations and it depends on the complexity of the preference model of the DM.

Figure 4: Group A of experiments (num. of ref. pts)



(a) 1 ref. pt (b) 2 ref. pts (c) 3 ref. pts

Figure 5: Group B of experiments (num. of ref. pts)



(a) 1 ref. pt (b) 2 ref. pts (c) 3 ref. pts

4.2.3 *Computation time*

The average computation time is provided in Table 5. We observed that, even for the case involving 3 reference points and 10 criteria, the computation time remains within around 3 minutes. These results should be further compared to the computation times of algorithms using MIP formulations.

Besides, not only should the runtime value be measured, but also the evolution of computation time as a function of the criteria number or the number of the reference points should be investigated. It is interesting while comparing with other MIP-based algorithms, as it demonstrates the significant runtime advantage of the metaheuristic algorithm.

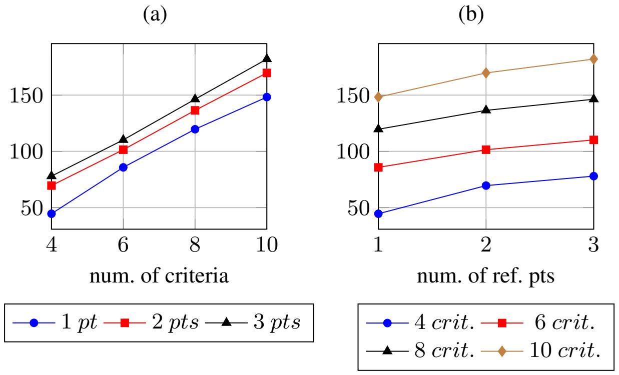
Table 4: Rapidity of convergence (group B, in number of iterations)

Num. Cri.	1 ref. pt	2 ref. pts	3 ref. pts
4	14	21	19
6	20	21	22
8	20	29	31
10	23	29	32

Table 5: Computation time (group B, in seconds)

Num. Cri.	1 ref. pt	2 ref. pts	3 ref. pts
4	44.55	69.58	77.98
6	85.78	101.48	110.20
8	119.67	136.46	146.35
10	148.28	169.75	182.06

Figure 6: Computation time (in seconds)



By drawing the runtime curves, as shown in Figure 6(a), we notice that the computation time is proportional with the criteria number. We remind that it increases exponentially in the MIP-based S-RMP disaggregation methods (Section 2.3.2). It endues the metaheuristic approach with the capacity of dealing with the decision instances that involve a large quantity of evaluating criteria.

Besides, we also notice that we consume more computation time when dealing with more reference points, but not as much as for the LP-based methods. To be clear, we draw once again the runtime curves in function of the number of reference points (as shown in Figure 6(b)). At this point of view, the obtained curves are all sub-linear regardless of the criteria number while in the LP-based S-RMP disaggregation methods, they are usually exponential vs. the number of reference points.

4.3 Discussion

By checking the result of the numerical tests, we notice that the preset number of reference points should be carefully adjusted to derive a better solution, since the final RA is intensively reduced (except for 10.3.A) in Group A (i.e. learning S-RMP models with single reference point) of experiments when the initial number of reference points increases. Nevertheless, starting from an S-RMP model with single reference point shows greater interpretability to the problem. It actually expresses a very natural rule based on the distinction of two classes of evaluation on each criterion. Besides, it helps us to estimate the complexity of S-RMP models and to fix the number of reference points. It means that we should consider increasing the number of reference points when the final RA is not as satisfactory as we expected.

Moreover, we also notice that the number of criteria should be limited. On one hand, we observe that, when the number of reference points is fixed, the final RA is descending progressively when the criteria number increases. On the other hand, the exceptional phenomenon that we observed (10.3.A) shows that, when there are more criteria, it is easier to derive an S-RMP model that involves only one reference point and gives a better RA. In other words, it is easier to adjust one reference point by calibrating its valuation and the weights of criteria than to calibrating multiple reference points at the same time.

The study about the rapidity of convergence of the algorithm could help us to adjust when to interrupt the algorithm when dealing with real datasets, and to economize the total computation time in different application circumstances. For instance, in the online web applications, the response time should usually be much more valuable than in the offline recommendations. To obtain the most accurate result, it is possible to run the program during a very long time, even day and night, in the offline cases. However, an acceptable solution should be worked out as quickly as possible in the online cases and then be adjusted step by step by a follow-up interactive process.

5 Conclusion

This paper presents an efficient metaheuristic approach to infer S-RMP models from a large set of pairwise comparisons provided by the DM. The proposed algorithm was tested with a large quantity of artificially generated data that simulates a variety of different decision circumstances. Firstly, the metaheuristic is able to deal with instances involving as many as 500 pairwise comparisons. Suchlike instances cannot be solved using MIP formulations. Secondly, the computation time is proportional with the number of criteria involved and sub-linearly increases with the number of reference points. Finally, even if the metaheuristic is not able to learn an S-RMP model which to restore all pairwise comparisons, it infers S-RMP models which restore up to at least 95% of the input information within a reasonable computation time. We remark that it constitutes a good trade-off of quality of result vs. computation time.

One of the interesting questions emerged from this work is the adjustment of the number of reference points. Moreover, the performance of the proposed metaheuristic should also be explored when we are in presence of inconsistencies, for example, in the case of group decision problems or in the case of real world applications.

REFERENCES

[1] Ilhem Boussaïd, Julien Lepagnot, and Patrick Siarry, 'A survey on optimization metaheuristics', *Information Sciences*, **237**, 82–117, (2013).

[2] Denis Bouyssou and Thierry Marchant, 'Multiattribute preference models with reference points', *European Journal of Operational Research*, **229**(2), 470 – 481, (2013).

[3] John Butler, Jianmin Jia, and James Dyer, 'Simulation techniques for the sensitivity analysis of multi-criteria decision models', *European Journal of Operational Research*, **103**(3), 531–546, (1997).

[4] Olivier Cailloux, Patrick Meyer, and Vincent Mousseau, 'Eliciting ELECTRE TRI category limits for a group of decision makers', *European Journal of Operational Research*, **223**(1), 133–140, (2012).

[5] Kenneth A. De Jong, 'Genetic algorithms are not function optimizers.', in *FOGA*, pp. 5–17, (1992).

[6] Michael Doumpos, Yannis Marinakis, Magdalene Marinaki, and Constantin Zopounidis, 'An evolutionary approach to construction of outranking models for multicriteria classification: The case of the ELEC-

TRE TRI method', *European Journal of Operational Research*, **199**(2), 496–505, (2009).

[7] Michael Doumpos and Constantin Zopounidis, 'On the development of an outranking relation for ordinal classification problems: An experimental investigation of a new methodology', *Optimization Methods and Software*, **17**(2), 293–317, (2002).

[8] Michael Doumpos and Constantin Zopounidis, 'Preference disaggregation and statistical learning for multicriteria decision support: a review', *European Journal of Operational Research*, **209**(3), 203–214, (2011).

[9] Johannes Fürnkranz and Eyke Hüllermeier, *Preference learning*, Springer, 2010.

[10] Eric Jacquet-Lagrèze and Yannis Siskos, 'Preference disaggregation: 20 years of mcda experience', *European Journal of Operational Research*, **130**(2), 233–245, (2001).

[11] Agnès Leroy, Vincent Mousseau, and Marc Pirlot, 'Learning the parameters of a multiple criteria sorting method', in *Algorithmic Decision Theory*, eds., RonenI. Brafman, FredS. Roberts, and Alexis Tsoukiàs, volume 6992 of *Lecture Notes in Computer Science*, 219–233, Springer Berlin Heidelberg, (2011).

[12] Jinyan Liu, Vincent Mousseau, and Wassila Ouerdane, 'Preference elicitation from inconsistent pairwise comparisons for multi-criteria ranking with multiple reference points', *ICISO 2013*, 120, (2013).

[13] Vincent Mousseau, 'Eliciting information concerning the relative importance of criteria', in *Advances in Multicriteria Analysis*, eds., PanosM. Pardalos, Yannis Siskos, and Constantin Zopounidis, volume 5 of *Nonconvex Optimization and Its Applications*, 17–43, Springer US, (1995).

[14] Vincent Mousseau and Roman Slowinski, 'Inferring an electre tri model from assignment examples', *Journal of global optimization*, **12**(2), 157–174, (1998).

[15] Marc Pirlot, 'General local search methods', *European Journal of Operational Research*, **92**(3), 493–511, (1996).

[16] Antoine Rolland, *Procédures d'agrégation ordinale de préférences avec points de référence pour l'aide à la décision*, Ph.D. dissertation, Université Pierre et Marie Curie-Paris VI, 2008.

[17] Antoine Rolland, 'Reference-based preferences aggregation procedures in multi-criteria decision making', *European Journal of Operational Research*, **225**(3), 479 – 486, (2013).

[18] Bernard Roy, *Multicriteria methodology for decision aiding*, volume 12, Springer, 1996.

[19] Olivier Sobrie, Vincent Mousseau, and Marc Pirlot, 'Learning a majority rule model from large sets of assignment examples', in *Algorithmic Decision Theory*, eds., Patrice Perny, Marc Pirlot, and Alexis Tsoukiàs, volume 8176 of *Lecture Notes in Computer Science*, 336–350, Springer Berlin Heidelberg, (2013).

[20] Olivier Sobrie, Vincent Mousseau, and Marc Pirlot, 'A majority rule sorting model to deal with monotone learning sets', Technical report, Laboratoire Génie Industriel, Ecole Centrale Paris, (June 2014). Cahiers de recherche 2014-01.

[21] William M Spears, Kenneth A De Jong, Thomas Bäck, David B Fogel, and Hugo De Garis, 'An overview of evolutionary computation', in *Machine Learning: ECML-93*, pp. 442–459. Springer, (1993).

[22] Jun Zheng, *Preference Elicitation for Aggregation Models based on Reference Points: Algorithms and Procedures*, Ph.D. dissertation, Ecole Centrale Paris, 2012.

[23] Jun Zheng, Antoine Rolland, and Vincent Mousseau, 'Preference elicitation for a ranking method based on multiple reference profiles', Technical report, Laboratoire Génie Industriel, Ecole Centrale Paris, (August 2012). Cahiers de recherche 2012-05.

Inferring the parameters of a majority rule sorting model with vetoes on large datasets

Alexandru-Liviu Olteanu[1 2] and **Patrick Meyer** [12]

Abstract. When dealing with a majority rule sorting model, the standard approach to indirectly infer its parameters is to use an approach based on assignment examples. However, when the sets of assignments are large the use of exact approaches quickly becomes impractical. While at least another metaheuristic approach has already been proposed to deal with this issue, we present in this paper a related metaheuristic approach. We carefully compare the two approaches as well as extend the latter to deal with majority rule models which also contain vetoes. The approaches are compared using both constructed and real data.

1 Introduction

Multicriteria (MC) decision aiding is the activity which provides a decision maker (DM) with a prescription on a set of decision alternatives, when facing multiple, usually conflicting viewpoints. The DM, who is either a single person or a collegial body, takes the responsibility for the decision act and bears a value system or preferences related to the decision problem, which should be taken into account in the final prescription. The finite set A of decision alternatives represents the potential options on which the DM has to make a decision. The decisions on these alternatives are considered as difficult because multiple conflicting perspectives have to be considered. They are represented by a finite set J of criteria indexes. Usually, three types of decision problems are considered forward in this context [17]:

- the *choice problem* which aims to recommend a subset of alternatives, as restricted as possible, containing the "satisfactory" ones;
- the *sorting problem* which aims to assign each alternative into predefined categories or classes;
- the *ranking problem* which aims to order the alternatives by decreasing order of preferences.

In this article we focus on the second category of decision problems.

To support DMs facing a MC decision problem, various methodologies have been proposed [8, 17]. Roughly speaking, they originate from two methodological schools (outranking and value-based techniques). The main differences between these two streams of thoughts lie in the way the alternatives are compared and in the type of information which is required from the DM. Among other things, outranking methods might be preferable if the evaluation scales of the criteria are very heterogeneous and if the DM would like to model some impreciseness about his preferences in the model, whereas value-based methods can be favoured if the criteria are evaluated mostly on numerical scales and if a compensatory behaviour of the DM should be modeled. We choose to explore in this research a sorting technique which is based on the outranking paradigm. It is a simplified version of the Electre Tri [7, 14, 16] method, which can be used quite easily, as it does not need a lot of technical parameters to be tuned properly. The version considered here is very close to the version studied by [2, 3].

To model the preferences of the DM, this sorting technique requires criteria importance parameters, category limits separating the categories, as well as veto thresholds for each category limit and each criterion. These parameters can either be obtained directly from the DM (which in most practical situations is not realistic), or learned from assignment examples provided by the DM for each of the categories. As we will show in the sequel, most of these learning techniques are not appropriate if the number of assignment examples becomes large, as they use mathematical programming techniques involving binary variables to obtain these preferential parameters.

It has therefore been suggested by [18] to use a technique based on a metaheuristic to learn the parameters of the sorting model. We use similar ideas here by iteratively running a sequence of two steps. The first step infers the criteria importance weights and majority threshold while keeping the category limits fixed, whereas the second step does the opposite and also searches for veto thresholds. Compared to [18], first we do not employ a population of solutions but only a single one that evolves over time, second the heuristic for guiding the algorithm, although similar to that in [18], differs in the use of majority margins on two levels to guide a simulated annealing implementation, third contains several adaptive parameters which are used to steer the algorithm, and last, veto thresholds may also be learned. The inclusion of vetoes has also been studied in [19], although new veto definitions are studied and the inference approach is an exact one.

To validate our approach, we perform tests on artificially generated benchmarks, as well as classical ones involving real data [1].

The rest of the article is structured as follows. We first introduce the preference model in Section 2. Then, in Section 3 we detail the proposed approach, before validating it on benchmarks in Section 4. Finally, in Section 5 we present some conclusions and perspectives for future work.

2 The preference model

As mentioned earlier, we consider here a simplified version of the Electre Tri [7, 14, 16] method, which is appropriate for a lot of practical applications. It is close to the version axiomatized in [2, 3].

Electre Tri requires, as a definition of the preferences of a DM, criteria importance parameters, category limits separating the categories, as well as veto thresholds for each category limit and each cri-

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

[2] Université Européenne de Bretagne

terion. The criteria importance parameters include a weight for each of the criteria and a majority threshold that defines when a coalition of criteria is good enough to be decisive. The category limits separate, for each criterion, two consecutive categories, and a veto threshold defines, for a given category limit and a given criterion, a performance which is too bad for the alternative to be assigned to the given category.

Consider a finite set of alternatives A, a set of category limits $B = \{b_1, \dots b_k\}$, and a finite set of criteria indexes J. A criterion g_j ($j \in J$) is a function from $A \cup B$ to $\mathbb{R}$ where $g_j(a)$ denotes the performance of the alternative a on criterion g_j. The alternatives have to be sorted in k categories, $c_1, \dots, c_k$, ordered by their desirability. c_1 is the worst category, and c_k is the best one. Each category c_h is defined by the performances of its lower frontier, or category limit, b_{h-1} and its upper frontier b_h of B (except the worst category c_1 has no lower frontier). The performances are here supposed to be such that a higher value denotes a better performance and the performances on the frontiers are non-decreasing, i.e. $\forall j \in J, 2 \leq h \leq k : g_j(b_{h-1}) \leq g_j(b_h)$.

To sort the alternatives, Electre Tri uses the concept of outranking relation. The assignment rule used here, known as the pessimistic rule, assigns an alternative a to the highest possible category c_h such that the alternative outranks the category's lower frontier b_{h-1}. An alternative a outranks a frontier b_{h-1} if and only if there is a sufficient coalition of criteria supporting the assertion "a is at least as good as b_{h-1}", and no criterion strongly opposes (vetoes) that assertion. To compute this, preferential parameters, representing the DM's preferences, are used. The coalition of criteria in favour of the outranking, $\forall a \in A, 1 \leq h \leq k$, is defined as

$$\sum_{j \in J} w_j C_j(a, b_{h-1}), \tag{1}$$

where w_j is the weight of the criterion g_j, and $C_j(a, b_{h-1}) \in \{0, 1\}$ measures if a is at least as good as b_{h-1} from the point of view of the criterion j or not: $C_j(a, b_{h-1}) = 1 \Leftrightarrow g_j(a) \geq g_j(b_{h-1})$, 0 otherwise. The weights are defined so that they sum to one ($\sum_{j \in J} w_j = 1$). The coalition is compared to a majority threshold $\lambda \in [0.5, 1]$ extracted from the DM's preferences along with the weights. If $\sum_{j \in J} w_j C_j(a, b_{h-1}) < \lambda$, the coalition is not sufficient and the alternative does not outrank the frontier b_{h-1} and will therefore be assigned in category below c_h.

Even when the coalition is strong enough, a criterion may veto the outranking situation. It happens when $g_j(a) > v_j^{h-1}$. The veto threshold v_j^{h-1} is a value that the DM may define and represents the performance that, if not reached by some alternative a, forbids the alternative to be in category c_h. To summarize, alternative a outranks frontier b_{h-1} (and therefore is assigned to at least the category c_h) if and only if $\sum_{j \in J} w_j C_j(a, b_{h-1}) \geq \lambda$ and $\forall j \in J : g_j(a) > v_j^{h-1}$.

The weights and majority thresholds (defining the sufficient coalitions) and the category limits may be given directly by the DM. However, in practice, this requires that the DM understands how these values will be used. It is moreover a difficult process to directly ask the DM for these parameters. The approach used here supposes that he provides assignment examples which are used to infer the preferential parameters. We denote with $A^{'}$ a subset of the alternatives in A and with $K : A' \rightarrow \{1, ..., h\}$ the assignments of these alternative to the set of ordered classes.

Previous works aiming to infer preferential parameters for the Electre Tri procedure on the basis of assignment examples suggest either to find the entire Electre Tri preference model parameters [13] from assignment examples, or to find the importance coefficients only [12], or only the categories limits [15], the other parameters being supposedly known. Robust approaches are suggested which compute for each alternative a range of possible categories to which alternatives can be assigned under incomplete determination of the parameters [4, 5, 6]. Some tools deal with the problem of non existing preference model solutions which may arise because of an inconsistent set of assignment examples (i.e. assignment examples that do not match Electre Tri) [11, 10].

However, learning these parameters requires linear programming techniques which necessitate the use of binary variables. In our context, where potentially large sets of assignment examples are involved, such an approach cannot be considered, as it requires large computing times. Similarly as in [18], we suggest to use a technique based on a metaheuristic to learn the parameters of the sorting model. The next section presents this approach and compares its characteristics to those of [18].

3 Proposed approach

In order to overcome the difficulties raised by finding the parameters of the preference model in an exact manner, we propose the use of a hybrid approach combining a linear programming approach and a metaheuristic iteratively in a similar fashion to the approach of [18]. However, there are a number of differences between the two approaches, mainly in the overall structure of the approach, which does not employ a population of solutions but only a single one that evolves over time, as well as the inclusion of several adaptive parameters within the approach. The approach looks to maximize the classification accuracy of the model over a sample of alternatives $A^{'}$. The construction or choice of the elements that are included in the sample is not in the scope of this work.

Algorithm 1 Proposed approach;

```
Input: Initial solution $s_0$, Initial temperature $T_0$.
 1: $\lambda, w, b, v$ = InitializeModelParameters();
 2: $dT, rN$ = InitializeAlgorithmParameters();
 3: $best\_f$ = Fitness($\lambda, w, b, v$);
 4: while not StoppingCondition() do
 5:     /* Linear program for weights and majority threshold */
 6:     $\lambda^{'}, w^{'}$ = LP($b, v$);
 7:     /* Metaheuristic for category profiles and vetoes */
 8:     $b^{'}, v^{'}$ = MH($\lambda^{'}, w^{'}, b, v, dT$);
 9:     if $best\_f <$ Fitness($\lambda^{'}, w^{'}, b^{'}, v^{'}$) then
10:         $best\_f$ = Fitness($\lambda^{'}, w^{'}, b^{'}, v^{'}$);
11:         $\lambda, w, b, v = \lambda^{'}, w^{'}, b^{'}, v^{'}$;
12:     else if $restart == 0$ then
13:         $\lambda, w, b, v$ = InitializeModelParameters();
14:     $dT, rN$ = UpdateAlgorithmParameters();
Output: $\lambda, w, b, v$.
```

The proposed approach (illustrated in Algorithm 1) divides the original problem of finding all the parameters of the preference model into two sub-problems:

- **the LP step**: finding the majority threshold and the criteria weights while the category and veto profiles are fixed (Algorithm 1 line 6);
- **the MH step**: finding the category and veto profiles while the majority threshold and the criteria weights are fixed (Algorithm 1 line 8).

These two steps are iterated until the algorithm converges to a final, close to optimal, solution (as will be shown in Section 4) or until a given amount of time has passed.

The algorithm also has two parameters, dT and rN, which are used to improve its efficiency.

The first parameters, dT, is used inside the metaheuristic step and influences the number of iterations that this step performs. Initially dT is high, which leads to the metaheuritic to perform a low number of iterations. dT is decreased when the metaheuristic does not improve the best found classification accuracy and is increased otherwise. In this way the metaheuristic is not run pointlessly for too long when it is able to improve the solution in a lower number of iterations and it is given more time otherwise.

The second parameter, rN, corresponds to the maximum number of non-improving iterations of the main loop of the algorithm that are allowed before restarting from the initial solution. This parameter is usually fixed beforehand and is used to restart the algorithm in case it converges to a potentially non-optimal solution.

3.1 Initialization

The initialization step is used in order to set the starting values of all the parameters. At this time the majority threshold is set to 0.5, while the criteria are given equal importance in the form of equal weights. Additionally, all veto thresholds for the category profiles are set to the least preferred evaluations of the alternatives on each criterion making them initially inactive.

The category profiles are constructed using a greedy heuristic which considers each criterion independently from the rest and places the value of a profile b_h so that it separates as much as possible the values of the alternatives that are classified in categories above the profile and the values of the alternatives that are classified in categories below the profile:

$$max: \sum_{a \in A'} h_{\text{init}}(a,h), \forall j \in J, \forall h \in \{1,\dots,k-1\} \text{ ,where} \quad (2)$$

$$h_{\text{init}}(a,h) = \begin{cases} 1 & \text{, if } K(a) > h \text{ and } a_j \geqslant b_j^h \\ & \text{or } K(a) \leqslant h \text{ and } a_j < b_j^h; \\ 0 & \text{, otherwise.} \end{cases} \quad (3)$$

3.2 LP step

The linear program for the first step of the approach is presented in Figure 1.

The assignments of the alternatives and the fixed category profiles are used indirectly by the linear program in the form of the C^+ and C^- parameters. These parameters correspond to the “at least as good as” assertions on each criterion between an alternative in A' and the category profile delimiting the assigned category and the category above, respectively the category below.

Aside from the constraint on the criteria weights summing up to 1, the other two constraints are used in order to validate the assignment of each alternative to its given category. It may be noticed that when the α^2 variable is strictly positive, the considered alternative is not considered to be at least as good as the category profile delimiting the assigned category and the category below, therefore the alternative will be assigned to a lower category. Similarly when the β^2 variable is strictly positive, the considered alternative is considered to be at least as good as the category profile delimiting the assigned category and the category above, therefore the alternative will be assigned to a higher category. The objective function tries to minimize these two variables, therefore aiming at minimizing the misclassifications. In order to keep the mathematical program simple, these variables are not binary, therefore the objective function does not directly minimize the number of misclassifications. This is an accepted trade-off in order to keep the first step tractable.

Figure 1. Linear program for the first step;

Parameters:

$$\begin{array}{ll} A', J & \\ C^+(a,j) \in [0,1] & \forall a \in A', \forall j \in J \\ C^-(a,j) \in [0,1] & \forall a \in A', \forall j \in J \\ \gamma \in]0,1[& \end{array}$$

Variables:

$$\begin{array}{ll} \lambda \in [0.5,1] & \\ w_j \in [0,1] & \forall j \in J \\ \alpha^1(a), \alpha^2(a), \beta^1(a), \beta^2(a) & \forall a \in A' \end{array}$$

objective:

$$min \sum_{a \in A'} (\alpha^2(a) + \beta^2(a))$$

Constraints:

$$\begin{array}{lll} s.t. & \sum_{j \in J} w_j = 1 & \\ & \sum_{j \in J} [C^-(a,j) \cdot w(j)] - \alpha^1(a) + \alpha^2(a) = \lambda & \forall a \in A' \\ & \sum_{j \in J} [C^+(a,j) \cdot w(j)] + \beta^1(a) - \beta^2(a) + \gamma = \lambda & \forall a \in A' \end{array}$$

As the the category profiles and their veto thresholds are fixed, it should also be noted that any alternative in A' that is in a veto situation with either the upper or lower profile of the category to which it should be assigned is not included in the linear program as no change in the criteria weights and majority threshold could impact its final classification.

3.3 MH step

The second step of the approach consists in a slight adaptation of the simulated annealing algorithm [9].

Algorithm 2 Simulated annealing;

```
Input: Initial temperature $T_0$, Temperature decrease parameter $dT$.
T = T_0;
while $T > 0$ do
    for all $j \in J$ do
        for all $h \in \{1,\dots,k-1\}$ do
            pick several $x \in [\min\{v_j^h, b_j^{h-1}\}, b^{h+1}]$ randomly;
            select $x$ which maximizes $H_b(h,j,x)$
            if $Heuristic(x) > 0$ or $random < e^{\frac{-1}{T}}$ then
                $b_j^h = x$
                $UpdateAssignments()$;
            pick several $x \in [v_j^{h-1}, \min\{v_j^{h+1}, b_j^h\}]$ randomly;
            select $x$ which maximizes $H_v(h,j,x)$
            if $Heuristic(x) > 0$ or $random < e^{\frac{-1}{T}}$ then
                $v_j^h = x$
                $UpdateAssignments()$;
    $T = T - dT$;
Output: $b, v$.
```

The simulated annealing algorithm performs changes to the category and veto profiles across several iterations. Each iteration is linked to a temperature parameter which decreases over time. In the beginning, at high temperatures, the algorithm may perform more frequently changes to the profiles which would lead to a decrease of the model fitness, while towards the end, as the temperature decreases, such changes get less frequent. Every iteration gives the opportunity of each profile to have each of its values on the set of criteria changed. Two heuristics H_b and H_v are used to determine the amount of increase or decrease in the fitness of the model given a new value on a criterion for a category profile, respectively a veto profile. In order to simplify them, we denote with A_h^l the set of alternatives that are classified by the model in class h while in the assignment examples they are placed in class l. Additionally, we define the flag V to indicate whether an alternative $a \in A$ has a lower evaluation on criterion $j \in J$ than a given veto profile:

$$V_j(a, v_h) = \begin{cases} 1 & \text{, if } g_j(a) < v_j^{h-1}; \\ 0 & \text{, otherwise.} \end{cases} \tag{4}$$

The two heuristics are listed in Equations (5), (6), (7), (8), (9) and (10).

$$H_b(h,j,x) = \sum_{a \in A} H_b^1(a,h,j,x) + \frac{1}{|A|} \sum_{a \in A} H_b^2(a,h,j,x). \tag{5}$$

$$H_b^1(a,h,j,x) = \begin{cases} +1 & \text{, if } (a \in A_h^{h+1} \text{ and } x > a_j \geqslant b_j^h \text{ and } \sum\limits_{i \in J} w_i C_i(a,b_h) - w_j < \lambda) \\ & \text{ or } (a \in A_{h+1}^h \text{ and } \sum\limits_{i \in J} V_i(a,v_h) = 0 \text{ and } b_j^h > a_j \geqslant x \text{ and } \sum\limits_{i \in J} w_i C_i(a,b_h) + w_j \geqslant \lambda); \\ -1 & \text{, if } (a \in A_h^h \text{ and } \sum\limits_{i \in J} V_i(a,v_h) = 0 \text{ and } b_j^h > a_j \geqslant x \text{ and } \sum\limits_{i \in J} w_i C_i(a,b_h) + w_j \geqslant \lambda) \\ & \text{ or } (a \in A_{h+1}^{h+1} \text{ and } x > a_j \geqslant b_j^h \text{ and } \sum\limits_{i \in J} w_i C_i(a,b_h) - w_j < \lambda). \end{cases} \tag{6}$$

$$H_b^2(a,h,j,x) = \begin{cases} +1 & \text{, if } (a \in A_h^{h+1} \text{ and } x > a_j \geqslant b_j^h \text{ and } \sum\limits_{i \in J} w_i C_i(a,b_h) - w_j \geqslant \lambda) \\ & \text{ or } (a \in A_{h+1}^h \text{ and } \sum\limits_{i \in J} V_i(a,v_h) = 0 \text{ and } x > a_j \geqslant b_j^h); \\ -1 & \text{, if } (a \in A_h^{h+1} \text{ and } b_j^h > a_j \geqslant x) \\ & \text{ or } (a \in A_{h+1}^h \text{ and } \sum\limits_{i \in J} V_i(a,v_h) = 0 \text{ and } x > a_j \geqslant b_j^h). \end{cases} \tag{7}$$

$$H_v(h,j,x) = \sum_{a \in A} H_v^1(a,h,j,x) + \frac{1}{|A|} \sum_{a \in A} H_v^2(a,h,j,x). \tag{8}$$

$$H_v^1(a,h,j,x) = \begin{cases} +1 & \text{, if } (a \in A_h^{h+1} \text{ and } x > a_j \geqslant v_j^h) \\ & \text{ or } (a \in A_{h+1}^h \text{ and } V_j(a,v_h) = 1 \text{ and } \sum\limits_{i \in J} V_i(a,v_h) = 1 \text{ and } v_j^h > a_j \geqslant x \text{ and } \sum\limits_{i \in J} w_i C_i(a,b_h) \geqslant \lambda); \\ -1 & \text{, if } (a \in A_h^h \text{ and } V_j(a,v_h) = 1 \text{ and } \sum\limits_{i \in J} V_i(a,v_h) = 1 \text{ and } v_j^h > a_j \geqslant x \text{ and } \sum\limits_{i \in J} w_i C_i(a,b_h) \geqslant \lambda) \\ & \text{ or } (a \in A_{h+1}^{h+1} \text{ and } x > a_j \geqslant v_j^h). \end{cases} \tag{9}$$

$$H_v^2(a,h,j,x) = \begin{cases} +1 & \text{, if } (a \in A_{h+1}^h \text{ and } \sum\limits_{i \in J} V_i(a,v_h) > 1 \text{ and } v_j^h > a_j \geqslant x) \\ & \text{ or } (a \in A_{h+1}^h \text{ and } V_j(a,v_h) = 1 \text{ and } \sum\limits_{i \in J} V_i(a,v_h) = 1 \text{ and } v_j^h > a_j \geqslant x \text{ and } \sum\limits_{i \in J} w_i C_i(a,b_h) < \lambda) \\ & \text{ or } (a \in A_h^h \text{ and } \sum\limits_{i \in J} V_i(a,v_h) = 0 \text{ and } v_j^h > a_j \geqslant x) \\ & \text{ or } (a \in A_h^h \text{ and } \sum\limits_{i \in J} V_i(a,v_h) > 1 \text{ and } v_j^h > a_j \geqslant x); \\ -1 & \text{, if } (a \in A_{h+1}^h \text{ and } x > a_j \geqslant v_j^h) \\ & \text{ or } (a \in A_h^h \text{ and } x > a_j \geqslant v_j^h). \end{cases} \tag{10}$$

Each heuristic contains two terms. The first is used to reflect immediate changes in classification accuracy due to the change of the profile evaluation on criterion j to the new value x, while the second is used to reflect potential changes in classification accuracy due to future changes of the profiles on other criteria. The second term is weighted so that it is dominated by the first. In this way, if the change in evaluation of the profile leads to an immediate change in classification accuracy, either positively or negatively, the second term does not interfere with it.

The first term from the heuristic for modifying a category profile adds the number of alternatives that are currently misclassified but which will become correctly classified as a result of the profile change, and subtracts the number of alternatives for which the reverse statement is valid. The resulting majority margin indicates through its sign whether the change in the profile evaluation leads to an increase in classification accuracy ($\sum\limits_{a \in A} H_b^1(a,h,j,x) > 0$), a decrease ($\sum\limits_{a \in A} H_b^1(a,h,j,x) < 0$), or to no change at all ($\sum\limits_{a \in A} H_b^1(a,h,j,x) = 0$). The second term works in a similar way, only that it adds the number of alternatives that are misclassified and that will remain misclassified as a result of the profile change, but for which the coalition of criteria in favour of this misclassification gets weakened, and it subtracts the number of such alternatives for which the coalition of criteria in favour of the misclassification gets strengthened. The alternatives that have been misclassified due to a veto are not taken into account as only a change to the veto profiles would allow for a change in assignment.

The second heuristic works in a similar way to the first only that it considers changes to the veto profiles. The exact rules may be deduced from its formula.

4 Results and validation

In this section we present several results of the proposed approach on both generated and real datasets and how it compares to the algorithm of [18].

In order to validate the approach we have constructed a series of benchmarks containing between 100, 1000 and 10000 alternatives defined on 10 criteria. On each benchmark we have used a model containing two categories, equally important criteria and a majority threshold of 50%. The only category delimiting profile was placed at the 50% level on each criterion, while the veto thresholds were placed at 20%. Alternatives were generated randomly and placed into one category or another based on the previously mentioned model. The number of alternatives in each category was set to the same value. The alternatives in the top category were generated in such a way as to contain only evaluations above the veto thresholds on all criteria and evaluations above the category delimiting profile on at least 50% of criteria. Two strategies for generating the alternatives in the bottom category were employed. On the one hand, a set of alternatives were generated in such a way as to contain only evaluations above the veto thresholds on all criteria and evaluations above the category delimiting profile on less than 50% of criteria, while on the other hand, a second set of alternatives were generated in the same way as those in the top category, only that on a randomly selected criterion their evaluations were lowered below the veto threshold.

Two sets of 10 benchmark instances were constructed for each benchmark size. The first set populated the alternatives in the bottom category using only the first strategy, while the second constructed half of the alternatives in this category using the first strategy and half using the second. The results that follow correspond to 25 executions of each algorithm on each of the 10 benchmark instances.

In a first study we compare the effectiveness of our metaheuristic and compare it also to that of [18]. We additionally consider the inclusion of vetoes in the majority rule model and the potential increase in its expressiveness. We consider several instances of our approach and of the approach of [18] which we denote as follows:

- MR-SORT$^{\mathrm{OM}}$: our approach without veto thresholds, without restarts and with the number of iterations of the metaheuristic fixed to that of the algorithm from [18];
- MRv-SORT$^{\mathrm{OM}}$: same as MR-SORT$^{\mathrm{OM}}$ but considering models with veto thresholds;
- MR-SORT$_a^{\mathrm{OM}}$: same as MR-SORT$^{\mathrm{OM}}$ but with variable number of iterations of the metaheuristic, which is followed by a local search step;
- MRv-SORT$_a^{\mathrm{OM}}$: same as MR-SORT$_a^{\mathrm{OM}}$ but considering models with veto thresholds;
- MR-SORT$_1^{\mathrm{SMP}}$: the approach from [18] containing only one model;

In Figures 2,3,4,5,6 and 7 we present the results over the 6 sets of benchmarks containing 100, 1000 and 10000 alternatives and the two generation strategies. We recall that the benchmarks constructed using the first strategy should be easily modelled using a majority rule model without vetoes, while those constructed using the second strategy should be modelled better using a majority rule model with vetoes. In all experiments we have used the entire set of alternatives as a learning set. We are at this point only interested to determine how well the algorithms are able to fit the models over these benchmarks, while experiments using samples over these datasets will be performed in a future study.

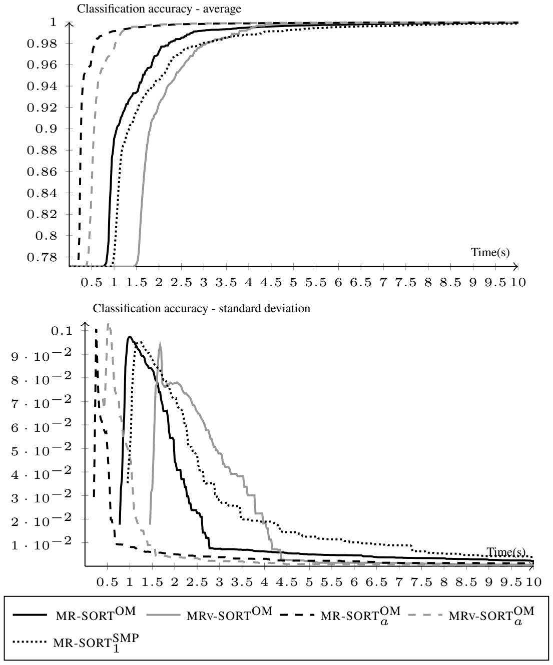


Figure 2. Average classification accuracy (top) and standard deviation (bottom) results over first set of benchmarks with 100 alternatives;

Starting with the results over the benchmarks containing 100 alternatives which have been generated specifically for a majority rule model without vetoes in Figure 2, we first notice that MRv-SORT$^{\mathrm{OM}}$ and MR-SORT$_1^{\mathrm{SMP}}$ start improving the initial solution at approximately the same time, therefore reflecting that the metaheuristics have indeed been tuned so that they perform the same number of iterations. We also quickly notice that MRv-SORT$^{\mathrm{OM}}$ takes roughly twice that amount of time to start, which is due to it having twice as many model parameters to tune. The adaptive versions of MR-SORT$^{\mathrm{OM}}$ and MRv-SORT$^{\mathrm{OM}}$ start much quicker due to their initial number of iterations of the metaheuristic being small. We observe that all versions of our algorithm perform well over these benchmarks. The non-adaptive versions appear to converge to the best solution slightly faster than MR-SORT$_1^{\mathrm{SMP}}$, which may be noticed both by looking at the average classification accuracy but also at the standard deviation from this value. MRv-SORT$^{\mathrm{OM}}$, while being slower to start, surpasses MR-SORT$^{\mathrm{OM}}$ in terms of robustness of the solution mid point during the experiment as seen by the standard deviation value. We consider this to be the effect of the veto construction which in certain cases forfeits the need to change the weights of the model in order to correctly classify certain alternatives. While these benchmarks have been constructed using a majority rule model without vetoes, it may be possible that vetoes could be used in order to correctly classify certain alternatives. Similar remarks may be found when considering the adaptive versions of our algorithm, however the appear to be much faster than the rest of the algorithms.

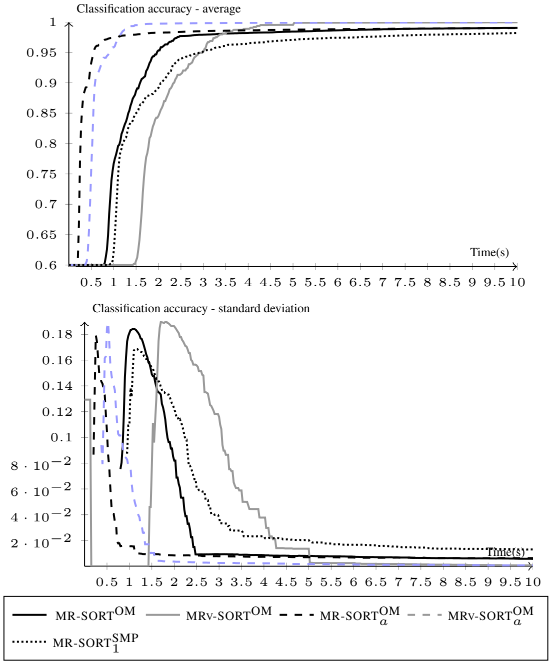


Figure 3. Average classification accuracy (top) and standard deviation (bottom) results over second set of benchmarks with 100 alternatives;

Looking at Figure 3 we find the results over the benchmarks containing 100 alternatives which have been generated for a majority rule model with vetoes. Most of the remarks from the first case are also valid in this case, however we notice that the algorithms using models with vetoes rise above the others. The difference however is small, which is due to the way in which the benchmarks have been constructed. It appears that constructing benchmarks where vetoes offer a big improvement over majority rule models without vetoes is a difficult task which we wish to explore further in the future.

Most of the remarks above hold for the results on the benchmarks containing 1000 and 10000 alternatives. We notice small discrepancies in the case of the benchmarks constructed using the first strategy. The algorithms start from a very good initial solution while the approaches using vetoes rise above the other quickly. While using the veto profiles has been previously seen to improve the solutions quickly, in these cases we do not see the expected rise of the other approaches to match the performance of these algorithms. Nevertheless, the difference may be due to the higher robustness of the algorithms accounting for vetoes, as seen through the standard deviation. Furthermore, as we are dealing with very good solutions, the probability of performing changes to them as to increase their fitness is rather low, which may explain the lack of increase in performance.

The results on the benchmarks constructed using the second strategy hold the same characteristics as the results on the smaller benchmarks, with the algorithms accounting for vetoes in the majority rule

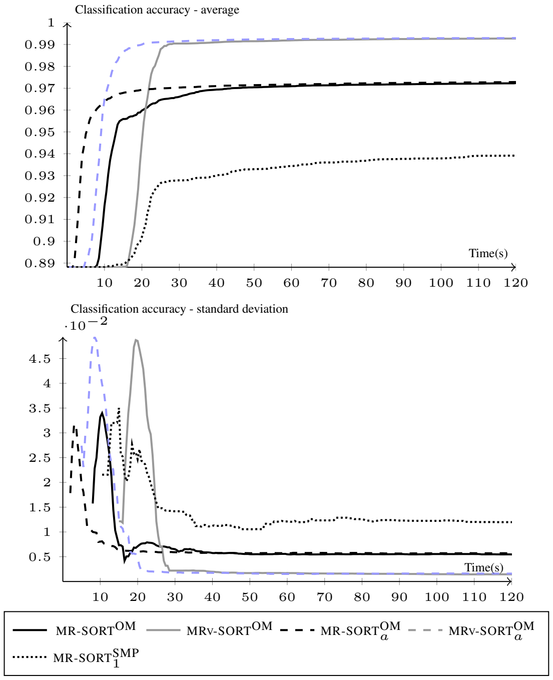


Figure 4. Average classification accuracy (top) and standard deviation (bottom) results over first set of benchmarks with $1,000$ alternatives;

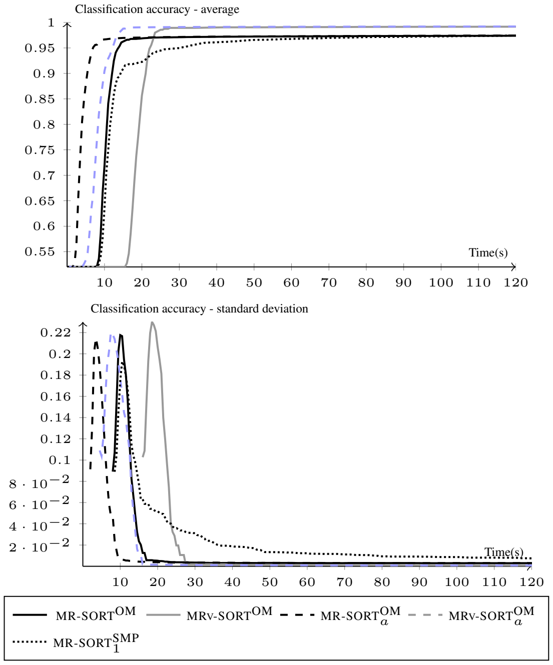


Figure 5. Average classification accuracy (top) and standard deviation (bottom) results over second set of benchmarks with $1,000$ alternatives;

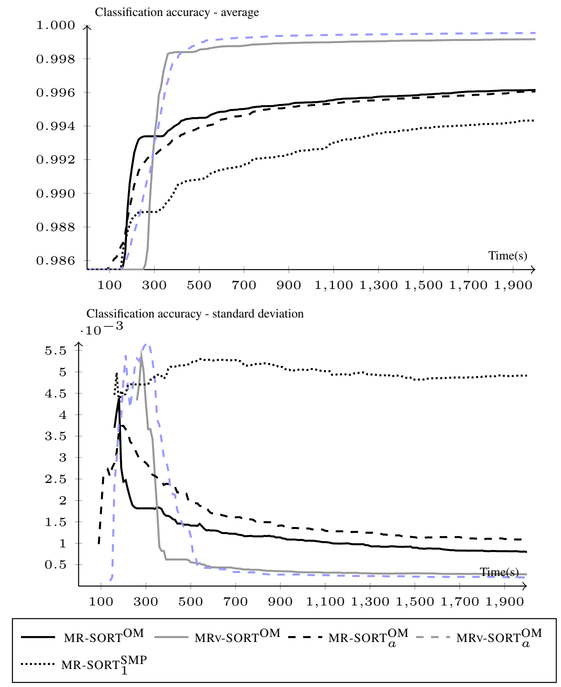


Figure 6. Average classification accuracy (top) and standard deviation (bottom) results over first set of benchmarks with $10,000$ alternatives;

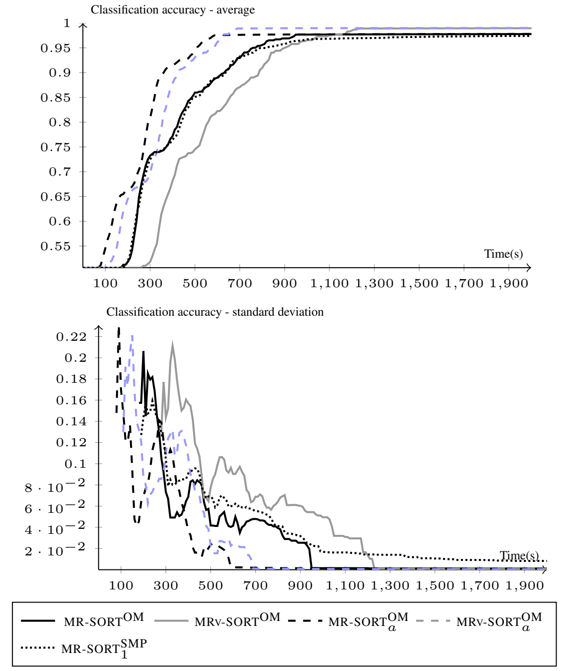


Figure 7. Average classification accuracy (top) and standard deviation (bottom) results over second set of benchmarks with $10,000$ alternatives;

model outperforming the others.

Overall, we may additionally observe that the algorithms scale linearly with the size of the dataset.

In a second study we compare the effectiveness of our approach compared to the standard form of the algorithm in [18]. We denote the approaches as follows:

- $\text{MR-SORT}^{\text{OM}}$: our approach without veto thresholds, without restarts and with the number of iterations of the metaheuristic fixed to that of the algorithm from [18];
- $\text{MR-SORT}^{\text{OM}}_5$: same as $\text{MR-SORT}^{\text{OM}}$ only that the model is reinitialized after 5 non-improving iterations;
- $\text{MR-SORT}^{\text{OM}}_5$: same as $\text{MR-SORT}^{\text{OM}}$ only that the model is reinitialized after 10 non-improving iterations;
- $\text{MR-SORT}^{\text{SMP}}_{10}$: the approach from [18] containing 10 models;

We have tested these approaches over several containing real data which have been taken from [1]. A description of a majority of these datasets may also be found in [20]. The algorithms have been executed 25 times over each dataset for an amount of time that would allow the $\text{MR-SORT}^{\text{SMP}}_{10}$ algorithm to perform at least 20 iterations, a number that is recommended by the authors of this approach.

In Table 1 we illustrate the average classification accuracy and standard deviation for the algorithms over each dataset.

Table 1. Results over the datasets in [1];

Dataset	Approach			
	$\text{MR-SORT}^{\text{OM}}$	$\text{MR-SORT}^{\text{OM}}_5$	$\text{MR-SORT}^{\text{OM}}_{10}$	$\text{MR-SORT}^{\text{SMP}}_{10}$
bcc	77.21 (0.57)	77.50 (0.44)	77.29 (0.56)	76.61 (0.00)
cpu	97.01 (0.24)	96.99 (0.29)	97.11 (0.31)	97.03 (0.47)
dbs	92.50 (0.53)	92.47 (0.60)	92.67 (0.65)	90.52 (0.81)
era	80.90 (0.22)	80.87 (0.19)	80.85 (0.16)	80.60 (0.34)
esl	91.74 (0.28)	91.79 (0.08)	91.74 (0.28)	90.70 (0.62)
lev	84.47 (0.04)	84.48 (0.04)	84.50 (0.01)	85.35 (0.77)
mmg	83.99 (0.87)	84.33 (0.76)	84.69 (0.80)	83.62 (0.28)
mpg	83.69 (0.10)	83.69 (0.10)	83.68 (0.05)	83.45 (0.38)

We observe that the $\text{MR-SORT}^{\text{OM}}$ approaches are generally slightly outperforming the $\text{MR-SORT}^{\text{SMP}}$ approach with a couple of exceptions for the cpu and lev datasets. In the case of the latter we have performed additional tests and reached the conclusion that the heuristics used by both algorithms are not able to guide the algorithms to the better solutions. In fact, the complete removal of the heuristic in our approach allowed us to reach the exact same values for the classification accuracy as the approach of $\text{MR-SORT}^{\text{SMP}}$.

Considering the variants of $\text{MR-SORT}^{\text{OM}}$ that restart the algorithm after several non-improving iterations, we may conclude that using this strategy is generally beneficial, as it either increases the average found classification accuracy or decreases its standard deviation, however further studies should be performed in order to find a good strategy of fixing the time when a model should be reinitialized.

5 Conclusion

In this work we have presented an approach for inferring the parameters of a majority rule sorting model with the potential inclusion of veto thresholds. While another metaheuristic approach has been previously proposed in [18], the main difference lies in the inclusion of veto thresholds in the model. Furthermore, our approach uses a single model and not a population of models, and contains a simulated annealing implementation at its core instead of a random search. We have also extended the method to adapt the time spent on the metaheuristic step as well as reinitialize the solution when it is not able to improve it. The presented approach has been tested and compared to the approach from [18] over a set of constructed benchmarks, as well as over datasets containing real data.

Future work should focus on studying strategies for adapting the time spend on the metaheuristic step and the reinitialization strategy in order to further improve the performance of the algorithm. We would also like to look into constructing benchmarks where the veto component of the majority rule model is more prominent and which would highlight the difference between a majority rule mode with vetoes and one without. Furthermore, we wish to consider the performance of the algorithm and that of a majority rule model with vetoes when the assignment examples represent only a sample of the original data.

REFERENCES

[1] Monotone learning datasets. `http://www.uni-marburg.de/fb12/kebi/research/repository/monodata`. Accessed: 2014-09-15.

[2] D. Bouyssou and T. Marchant, ‘An axiomatic approach to noncompensatory sorting methods in MCDM, I: the case of two categories’, *European Journal of Operational Research*, **178**(1), 217–245, (April 2007).

[3] D. Bouyssou and T. Marchant, ‘An axiomatic approach to noncompensatory sorting methods in MCDM, II: more than two categories’, *European Journal of Operational Research*, **178**(1), 246–276, (April 2007).

[4] L.C. Dias and J.N. Clmaco, ‘On computing ELECTRE’s credibility indices under partial information’, *Journal of Multi-Criteria Decision Analysis*, **8**(2), 74–92, (1999).

[5] L.C. Dias and J.N. Clmaco, ‘ELECTRE TRI for groups with imprecise information on parameter values’, *Group Decision and Negotiation*, **9**(5), 355–377, (September 2000).

[6] L.C. Dias, V. Mousseau, J. Figueira, and J.N. Clmaco, ‘An aggregation/disaggregation approach to obtain robust conclusions with ELECTRE TRI’, *European Journal of Operational Research*, **138**(2), 332–348, (April 2002).

[7] J. Figueira, V. Mousseau, and B. Roy, ‘ELECTRE methods’, in *Multiple Criteria Decision Analysis: State of the Art Surveys*, eds., J. Figueira, S. Greco, and M. Ehrgott, 133–162, Springer Verlag, Boston, Dordrecht, London, (2005).

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

[9] S. Kirkpatrick, C. Gelatt, and M. Vecchi, ‘Optimization by simulated annealing’, *Science*, **220**(4598), 671–680, (1983).

[10] V. Mousseau, L.C. Dias, and J. Figueira, ‘Dealing with inconsistent judgments in multiple criteria sorting models’, *4OR*, **4**(3), 145–158, (2006).

[11] V. Mousseau, L.C. Dias, J. Figueira, C. Gomes, and J.N. Clmaco, ‘Resolving inconsistencies among constraints on the parameters of an MCDA model’, *European Journal of Operational Research*, **147**(1), 72–93, (2003).

[12] 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, (April 2001).

[13] V. Mousseau and R. Slowinski, ‘Inferring an ELECTRE TRI model from assignment examples’, *Journal of Global Optimization*, **12**(2), 157–174, (1998).

[14] V. Mousseau, R. Slowinski, and P. Zielniewicz, ‘A user-oriented implementation of the ELECTRE TRI method integrating preference elicitation support’, *Computers & Operations Research*, **27**(7-8), 757–777, (2000).

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

[16] B. Roy, ‘The outranking approach and the foundations of ELECTRE methods’, *Theory and Decision*, **31**, 49–73, (1991).

[17] B. Roy, *Multicriteria Methodology for Decision Aiding*, Kluwer Academic, Dordrecht, 1996.

[18] O. Sobrie, V. Mousseau, and M. Pirlot, ‘Learning a majority rule model from large sets of assignment examples’, in *Algorithmic Decision Theory*, pp. 336–350. Springer Berlin Heidelberg, (2013).

[19] O. Sobrie, M. Pirlot, and V. Mousseau, ‘New veto relations for sorting models’, Technical report, Laboratoire Gnie Industriel, Ecole Centrale Paris, (October 2014). Cahiers de recherche 2014-04.

[20] A. Tehrani, W. Cheng, and E. Hüllermeier, ‘Preference learning using the Choquet integral: the case of multipartite ranking’, in *Proceedings of the 20th Workshop Computational Intelligence*, eds., Frank Hoffmann and Eyke Hüllermeier, pp. 119–130, Dortmund, Germany, (December 2010). KIT Scientific Publishing.

A Dataset Repository for Benchmark in MCDA

Antoine Rolland [1] and **Thi-Minh-Thuy Tran** [2]

Abstract. Several methods have been proposed in the past decades to deal with Multicriteria Decision Aiding (MCDA) problems. However, a comparison between these methods is always arduous as there is no benchmark in this domain. In the same time, people proposing new MCDA methods have no standardized data to deal with to prove the interest of their methods. We propose the creation of a web MCDA DataSet Repository to face this lack of data. This dataset repository is designed to be used by any multicriteria method, but particularly in the the frame of the DIVIZ platform. We detail the presentation of this repository in this paper. The dataset repository is available at http://data.decision-deck.org/

1 Introduction

MCDA aims at helping a Decision Maker to take decisions. Many different models have been proposed since more than 50 years (see [6] or [3] for a survey), among others:

- utility-based approaches, using linear (MAUT [11], AHP [15]) or non-linear (Choquet integral [8]) aggregation functions
- outranking approaches, like ELECTRE [7] or PROMETHEE [4] methods
- mixed methods, like rule-based methods [9, 10] and others.

There is still a great increase of the number of very specific methods to be proposed. All these methods are always presented as very interesting and perfectly adapted to the situation. The fact is that it is very difficult to test and compare different methods described in the literature, as they often are dedicated to one specific situation. Therefore, there is a lack of testing set of data on which one can try the different methods. Several solutions have already been proposed to increase the possibility of benchmark between MCDA method. We can cite the Decision Deck project which proposes a unified standard for MCDA data [2], and a unified web services platform through DIVIZ [13]. We can cite also a companion paper [5] which aims at proposing a simulation method to generate MCDA fictitious data from real ones.

Some other dataset repository exist, but as far as we know none of them is about MCDA. We can cite, among others:

- the **UCI** Machine Learning Repository [1]. It is a good resource for classification problems, but the a huge number of variables and/or alternatives make them inappropriate for a MCDA approach. The datasets are dedicated to statistical or machine learning approaches.
- **PrefLib** [12] is a database specialized in preference aggregation as in voting theory. Data are lists of complete or incomplete orders or preorders, but only cardinal informations are provided.

This paper presents the MCDA DataSet Repository. In the next section, we will first motivate the creation of the repository. We then present the functionalities of the MCDA DataSet Repository in section 3. Section 4 is dedicated to the contain of the MCDA DataSet Repository .

2 Motivations

As pointed out in the introduction, our main motivation is to provide to the MCDA community a large set of multicriteria decision situations and data. Following [12], the main motivations for building a case library are benchmarking, competition, realism, challenges and insularity. We agree on these motivations and precise some of them into the MCDA framework.

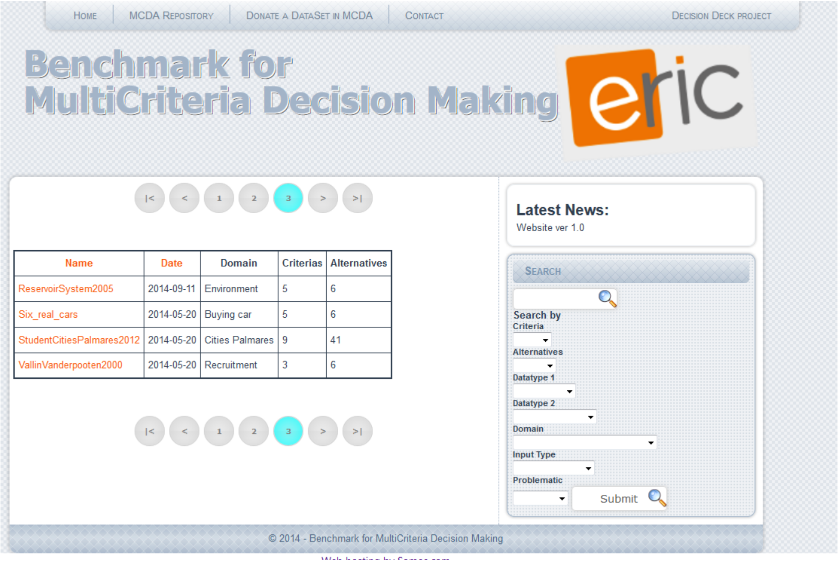
- **Benchmarking**: our main motivation is to provide to the community a large possibility to compare the results of several different methods on the same dataset. The use of XMCDA as an unified data standard should increase these possibility. The aim is not to try to prove the superiority of a specific method, but to help the analysis of the convergences and divergences of the different methods on a given dataset. Even if the theoretical divergences are now well known [3], it is interesting to see in practice on real data sets when the different methods give the same results or not.
- **Insularity** is defined in [12] as "most people work on their own problems and their own data". The use of a common problems repository should favour interactions and cross-over fertilization between different MCDA researchers.
- **Realism**: several variants and improvements of MCDA methods are based on the study of more and more specific cases. For example, the MAUT started with linear additive utilities. Then non-linear utilities have been proposed, and later non-additive aggregation functions have been introduced, increasing both the description capacities of the proposed methods and their complexity. Many of these improvements/complexifications are justified by ad-hoc examples. The use of datasets with real data should enable to justify (or not) the use of these improvements in practice.

In order to reach theses objectives, the datasets proposed into MCDA DataSet Repository should fulfil some criteria:

- **Quantity**: the number of proposed datasets should be large enough for each user to find a suitable case for its experiments. We start with more than 15 different cases but we expect an quick increase of the number of available datasets.
- **Diversity**: the success of the MCDA DataSet Repository will also stand into its capacity to provide a great variety of data and situations. We have paid much attention to select various datasets

[1] Laboratoire ERIC, université LYON 2, email: antoine.rolland@univ-lyon2.fr (corresponding author)

[2] Laboratoire ERIC, université LYON 1, email: thi-minh-thuy.tran@etu.univ-lyon1.fr



Name	Date	Domain	Criterias	Alternatives
ReservoirSystem2005	2014-09-11	Environment	5	6
Six_real_cars	2014-05-20	Buying car	5	6
StudentCitiesPalmares2012	2014-05-20	Cities Palmares	9	41
VallinVanderpooten2000	2014-05-20	Recruitment	3	6

Latest News:
Website ver 1.0

© 2014 - Benchmark for MultiCriteria Decision Making

Figure 1. MCDA DataSet Repository Homepage

into the first ones available on-line. Both real and fictitious data, qualitative, quantitative or mixed data are available.

- **Reliability**: it is essential that the provided datasets are free of bugs and mistakes. That's why we test carefully all the files contained into the proposed data sets. We then guaranty that all the data are coherent with the XMCDA standards and readable by the appropriate DIVIZ web services. We cannot of course be considered as responsible in case of misuse of the data.
- **Open access**: all the proposed data come from our private experiments or have been taken from the literature. We suppose that authors publishing their data are doing so for readers to be able to use them again, which is the spirit of MCDA DataSet Repository .

3 Functionalities

3.1 Access

The MCDA DataSet Repository is available with the following URL: `http://data.decision-deck.org/`

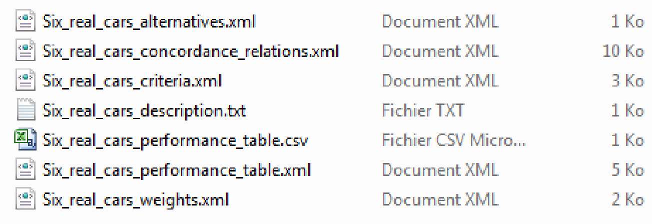
The repository homepage (see fig. 1) presents directly the different available datasets, and filters to help users to choose the dataset that better suits his/her needs.

3.2 Dataset structure

A dataset included in the MCDA DataSet Repository is composed of three types of files:

1. a **description** file, including a basic presentation of the data framework, a reference to the data source, and a brief description of the criteria and the alternatives.
2. a **performance table** file in a `.csv` format, making the performance data directly available on a line/column table. This table does not include any information about the type of data, scales and preference directions of the criteria and so on. Please refer to the description file for these informations.
3. as **many files** as needed in the `.xmcda` format (see [2] for details) to describe entirely the dataset. It includes at least a file for the alternatives, a file for the criteria and a file for the performance table. It can also include file with needed information for the use of specific methods like weights for a weighted mean, or importance weight for the use of ELECTRE methods for example.

All these files are jointed into a zip file in order to have the dataset loading facilitated (see fig. 2).



Six_real_cars_alternatives.xml	Document XML	1 Ko
Six_real_cars_concordance_relations.xml	Document XML	10 Ko
Six_real_cars_criteria.xml	Document XML	3 Ko
Six_real_cars_description.txt	Fichier TXT	1 Ko
Six_real_cars_performance_table.csv	Fichier CSV Micro...	1 Ko
Six_real_cars_performance_table.xml	Document XML	5 Ko
Six_real_cars_weights.xml	Document XML	2 Ko

Figure 2. MCDA DataSet Repository : a dataset

3.3 Filters

Facing the list of datasets, the user should like to access directly the cases that better suit its willing and needs. That's why the MCDA DataSet Repository proposes a filtering function (see fig. 3) including the following selection fields. Note that these fields can be jointly or independently selected.

- **Problematic**: inspired by the classification proposed by Roy [14], the dataset problematic can concern a ranking, sorting or choice problem. More problematic can be added if needed.
- **Data type** (1 & 2): it indicates whether the data are qualitative or quantitative, and if they are ordinal numbers, real numbers, fuzzy numbers or intervals.
- **Domain** specifies the application domain of the case study.
- **Inputs** indicates if the data are real data or fictitious (simulated) ones.
- The **number of criteria** should be a matter of interest to select the dataset, such as the following.
- The **number of alternatives** should also be a matter of interest, as some methods are dedicated to small, or large amounts of data.
- **Known Results** indicates if the result of the decision process has been specified in the source, i.e. which solution has been chosen in the case of a choice problem, or the final category affectation in the case of a sorting problem for example.

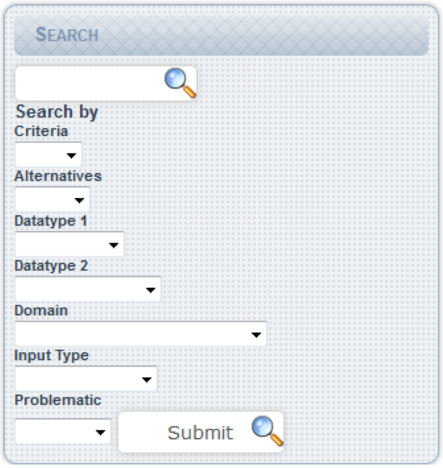


Figure 3. MCDA DataSet Repository : filtering

4 Contain

4.1 Content policy

The aim of the MCDA DataSet Repository is to propose as many datasets as possible. Any multicriteria decision-making situation is welcome in the MCDA DataSet Repository . Ideally, the MCDA DataSet Repository should contain both datasets coming from a real-life study cases and datasets with fictitious data specially built to illustrate a specific decision case.

The MCDA DataSet Repository can be supplied by several way:

- the authors still maintain a bibliography watch in order to supply the repository with data coming from case-studies proposed in the literature.
- any one can also propose a dataset. This can be easily done through the ”Donate a Dataset” button (see fig. 4)

Figure 4. MCDA DataSet Repository : donate a dataset

4.2 Example

Let us detail the dataset ”VallinVanderpooten2000”. This dataset is issued from [16]. The data are presented in the following table:

Performances	G1	G2	G3
A01	16	14	16
A02	10	18	12
A03	18	12	6
A04	18	4	20
A05	16	10	12
A06	6	14	18

The folder includes 6 files:

- **VallinVanderpooten2000_description.txt** contains a brief description of the dataset framework:

```
"In order to hire someone, a
recruitment agency evaluates 6
candidates through 3 tests:

- competences evaluation (G1)
- general knowledge evaluation (G2)
- motivation evaluation (G3)

These evaluations are noted on a [0;20]
scale and have to be maximized."
```

- **VallinVanderpooten2000_performanceTable.csv** contains the data in a `csv` format.

```
Performances;G1;G2;G3
A01;16;14;16
A02;10;18;12
A03;18;12;6
A04;18;4;20
A05;16;10;12
A06;6;14;18
```

- **VallinVanderpooten2000_alternative.xml** contains the alternatives list in a `xmcda` format.

```
<alternatives>
<alternative id="a01" />
<alternative id="a02" />
<alternative id="a03" />
<alternative id="a04" />
<alternative id="a05" />
<alternative id="a06" />
</alternatives>
</xmcda:XMCDA>
```

- **VallinVanderpooten2000_criteria.xml** contains the criteria list in a `xmcda` format.

```
<criteria>
<criterion id="G1" name="competences">
<scale>
<quantitative>
<preferenceDirection>max
</preferenceDirection>
<minimum><real>0</real></minimum>
<maximum><real>20</real></maximum>
</quantitative>
</scale>
<thresholds>
<threshold mcdaConcept="veto">
<constant>
<real>9</real>
</constant>
</threshold>
</thresholds>
</criterion>
(...)
</criteria>
```

- **VallinVanderpooten2000_performanceTable.xml** contains the performance table in a `xmcda` format.

```
<performanceTable id="normalised"
mcdaConcept="cardinalScales">
<alternativePerformances>
<alternativeID>a01</alternativeID>
<performance>
<criterionID>G1</criterionID>
<value> <real>16</real></value>
</performance>
<performance>
<criterionID>G2</criterionID>
<value> <real>14</real></value>
</performance>
<performance>
<criterionID>G3</criterionID>
<value><real>16</real></value>
</performance>
</alternativePerformances>
(...)
</performanceTable>
```

- **VallinVanderpooten2000_weight.xml** contains the criteria weights in a `xmcda` format.

```
<criteriaValues mcdaConcept="Importance"
name="significance">
<criterionValue>
<criterionID>G1</criterionID>
<value><real>0.6</real></value>
</criterionValue>
<criterionValue>
<criterionID>G2</criterionID>
<value><real>0.1</real></value>
</criterionValue>
<criterionValue>
<criterionID>G3</criterionID>
<value><real>0.3</real></value>
</criterionValue>
</criteriaValues>
```

5 Conclusion

In this paper we have introduced the first version of the MCDA DataSet Repository . The success of such a repository will come not only from the use of the proposed databased, but also from the contributions received from the research community. We then encourage all the MCDA practitioners to enrich the MCDA DataSet Repository through data donations. Please do not hesitate to contact us for any help.

REFERENCES

[1] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[2] Raymond Bisdorff, Patrick Meyer, and Thomas Veneziano, 'XMCDA : a standard XML encoding of MCDA data', in *EURO XXIII : European conference on Operational Research*, pp. 53 – 53, (2009).

[3] *Concepts and Methods of Decision-Making*, eds., Denis Bouyssou, Didier Dubois, Marc Pirlot, and Henri Prade, Wiley-ISTE, 2009.

[4] J.P. Brans and B. Mareschal, 'PROMETHEE methods', in *Multiple Criteria Decision Analysis: State of the Art Surveys*, eds., J. Figueira, S. Greco, and M. Ehrgott, 163–196, Springer Verlag, (2005).

[5] J. Cugliari, A. Rolland, and T.M.T. Tran, 'On the use of copula for mcda data simulation', in *DA2PL 2014 Workshop From Multiple Criteria Decision Aid to Preference Learning*, p. ??, (2014).

[6] *Multiple Criteria Decision Analysis: State of the Art Surveys*, eds., J. Figueira, S. Greco, and M. Ehrgott, Springer Verlag, Boston, Dordrecht, London, 2005.

[7] J. Figueira, V. Mousseau, and B. Roy, 'ELECTRE methods', in *Multiple Criteria Decision Analysis: State of the Art Surveys*, eds., J. Figueira, S. Greco, and M. Ehrgott, 133–53, Springer Verlag, Boston, Dordrecht, London, (2005).

[8] M. Grabisch, 'The application of fuzzy integrals in multicriteria decision making', *European Journal of Operational Research*, **89**, 445–456, (1996).

[9] S. Greco, B. Matarazzo, and R. Słowiński, 'Rough sets theory for multicriteria decision analysis', *European Journal of Operational Research*, **129**, 1–47, (2001).

[10] S. Greco, B. Matarazzo, and R. Słowiński, 'Rough sets methodology for sorting problems in presence of multiple attributes and criteria', *European Journal of Operational Research*, **138**, 247–259, (2002).

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

[12] Nicholas Mattei and Toby Walsh, 'Preflib: A library of preference data', in *Proceedings of Third International Conference on Algorithmic Decision Theory (ADT 2013)*, Lecture Notes in Artificial Intelligence. Springer, (2013).

[13] Patrick Meyer and Sébastien Bigaret, 'Diviz: A software for modeling, processing and sharing algorithmic workflows in MCDA', *Intelligent decision technologies*, **6**, 283 – 296, (2012).

[14] B. Roy, 'The outranking approach and the foundations of ELECTRE methods', in *Readings in Multiple Criteria Decision Aid*, ed., C.A Bana e Costa, 155–183, Springer-Verlag, Berlin, (1990).

[15] T.L. Saary, *The analytic hierarchy process*, McGraw Hill International, New York, 1980.

[16] P. Vallin and D. Vanderpooten, *Aide à la décision : une approche par les cas*, Ellipses, Paris, 2000. 2e édition, 2002.

Session 8

- Invited speaker: "*Preference modeling with Choquet integral*",
  Michel Grabisch, Université Paris 1
  In this talk, we show how capacities and the Choquet integral emerge as natural ingredients when building a multicriteria decision model, especially when the criteria cannot be considered as independent. To face the complexity of the model, we provide efficient submodels based on k-additive capacities, which are naturally connected with the interaction indices, quantifying the interaction existing among criteria in a group of criteria. The case of 2-additive capacities seems to be of particular interest, since it leads to a model which is convex combination of an additive model and max and min over any pair of two criteria. Lastly, we address the issue of the identification of the model through learning data and preferences.

15h30 Session 9

- "*Characterization of Scoring Rules with Distances: Application to Clustering of Rankings*", Paolo Viappiani, LIP6, Université Pierre et Marie Curie

- "*An interactive approach for multiple criteria selection problem*", Anil Kaya[1], Özgür Özpeynirci[1], Selin Özpeynirci[2],
  [1] Izmir University of Economics, Department of Logistics Management,
  [2] Izmir University of Economics, Industrial Engineering Department

- "*FlowSort parameters elicitation: the case of partial sorting*", Dimitri Van Assche, Yves De Smet,
  CoDE, Université libre de Bruxelles

- "*On confident outrankings with multiple criteria of uncertain significance*", Raymond Bisdorff, University of Luxemburg

Characterization of Scoring Rules with Distances: Application to Clustering of Rankings

Paolo Viappiani [1] [2]

Abstract. We consider the problem of clustering rank data, focusing on distance-based methods. Two main steps need to be performed: aggregating rankings of the same cluster into a representative ranking (the cluster's centroid) and assigning each ranking to its closest centroid according to some distance measure. A principled way is to specify a distance measure for rankings and then perform rank aggregation by explicitly minimizing this distance. But if we want to aggregate rankings in a specific way, perhaps using a scoring rule giving more importance to the first positions, which distance measure should we use?

Motivated by the (known) observation that the aggregated ranking minimizing the sum of the Spearman distance with a set of input rankings can be computed efficiently with the Borda rule, we build a taxonomy of aggregation measures and corresponding distance measures; in particular we consider extensions of Spearman that can give different weights to items and positions.

1 Introduction

It is often the case that data is available in the form of rankings (ordered lists of elements that express a preference order), for instance, this is the case of preference information obtained in electronic commerce applications. This paper deals with the problem of clustering preference data that is available in the form of rankings.

One motivation for clustering rankings is that by producing a (small) number of aggregated rankings we are able to provide a meaningful qualitative description for the entire population. In this work we focus on distance-based methods for clustering that are attractive because they do not make specific assumptions (contrary to probabilistic methods [2, 10], such as Mallows models, that assume a specific generative model for the rankings). A distance-based clustering method partitions the elements into clusters, so that the within-cluster distance is minimized; each cluster can then be associated with a representative element of the partition (the centroid). When considering rankings as elements to be clustered, the issue is to define a meaningful sound distance measure that can be used; in fact several alternative possibilities exist.

Another, related, problem is that of ranking aggregation. Several methods for aggregation of rankings have been proposed, notably in the social choice theory community. A motivation of this work is to study which aggregation methods can be formulated as the minimization of some distance measure between rankings. Distance-based clustering associates elements (in this case rankings of objects from the most to the least preferred) to clusters; in each cluster a representative ranking (centroid) is computed using some aggregation rule. Rankings are assigned to the cluster whose centroid is the closest.

When a distance measure is defined, a natural aggregation method is looking for the ranking that minimizes the sum of the distances with all input rankings. An interesting question is whether common aggregation methods have a corresponding distance that they minimize implicitly. If such connection can be made, clustering can often be made more efficient, as often ranking aggregation techniques are computationally less demanding than explicit minimization of distance.

2 Distance-based Clustering

We have a set of n items or objects and a set of m users or agents. A ranking π is a permutation on the set of available objects. Formally π is a function from $\{1,...,n\}$ to $\{1,...,n\}$ associating each item with its position (rank). As usual, the set of possible rankings is denoted as S_n. A ranking can be expressed alternatively in an explicit form of a tuple $\langle\pi^{-1}(1),...,\pi^{-1}(n)\rangle$, with $\pi^{-1}(r)$ being the r-th most preferred item; for example $\langle 2,1,3\rangle$ is the ranking for which item 2 is the most preferred, then item 1 is preferred, and finally item 3 is the least preferred (this corresponds to $\pi(1)=2$, $\pi(2)=1$ and $\pi(3)=3$).

In this paper we consider to have a number of rankings π_1,...,π_m, associated with different users or agents, and we want to partition them into different clusters. Let $f : i \rightarrow r$ assign rankings to clusters and d be a distance. Distance-based clustering is the problem of, given a number m of rankings, partitioning them in k clusters (or classes) so to minimize the within-cluster sum of distances with respect to some "central" rankings $\bar{\pi}^*_1, .., \bar{\pi}^*_k$ of each cluster.

$$(f^*, \bar{\pi}^*_1, .., \bar{\pi}^*_k) = \arg \min_{f,\bar{\pi}^*_1,...,\bar{\pi}^*_k} \sum_{z=1}^{k}\sum_{j=1}^{m} d(\bar{\pi}_z, \pi_j) \qquad (1)$$

The problem of clustering is frequently tackled in the literature with an iterative algorithm that proceeds in two steps. In the *assignment* step, each observation is assigned to the cluster whose "mean" yields the least within-cluster distance. In the *update* step, we calculate the new means to be the centroids of the observations in the new clusters. When items are vectors and the Euclidean distance is used, the problem is that of k-means clustering and the iterative algorithm described above is often called as well k-means [3].

Here, following [8] we adopt the same idea for clustering a set of rankings. We proceed in a iterative way. As in traditional k-means, we maintain a set of centroids (initialized randomly) and we associate each ranking with the cluster whose centroid is closest. Then,

[1] Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6

[2] CNRS, UMR 7606, LIP6, 4 Place Jussieu, 75005 Paris, France; email: paolo.viappiani@lip6.fr

[3] Also sometimes called Lloyd's algorithm.

we recompute the centroids for each of the clusters. We alternate between these two actions until further application of this methodology does not change the clusters anymore. The algorithm assumes that a suitable distance d on rankings is given. The algorithm's pseudocode is outlined below (Algorithm 1).

```
Algorithm 1: Distance-based clustering of rankings.
Data: $\pi_1, ..., \pi_m$ (population of rankings given by $m$ users), $k$
      (number of clusters)
Randomly initialize the centroids $\bar{\pi}_1, ..., \bar{\pi}_k$ ;
while there are changes in cluster assignments do
    Assign each ranking $\pi$ to the cluster whose distance to the
    centroid is lower
    $f(\pi_i) := \arg\min_{r=1,..,k} d(\pi_i, \bar{\pi}^r) \;\; i=1,..,m$;
    Find the centroid of each cluster
    $\bar{\pi}^r := \min_{\pi \in S_n} \sum_{\pi_t : f(\pi_t)=r} d(\pi, \pi_t) \;\; r = 1, ..., k$;
end
```

A well known fact is the following:

Observation 1. *Algorithm 1 converges to a local optimum.*

This observation is easily proved by showing that both main steps of the algorithm cannot increase the total distances of each data point to the centroid of its cluster. Since the distance-based clustering approach returns a local optimum, the algorithm may be run for a number of times (typically 10 or 20) and store the clustering assignment associated with the lowest sum of distances to the closest centroid.

Algorithm 1 requires the specification of a suitable distance measure. Rankings are particular "objects", and there are many different ways to define a distance between two rankings; some common distances are reviewed below in Section 3.2. Note however, that rank aggregation is often treated as a separate problem, especially in social choice literature, notably in voting methods (an aggregate ranking is obtained without considering an underlying distance measure).

The main contribution of this paper is to study the connection between rank aggregation (in particular, scoring rules) and ranking distances. We will propose new distances that allow to assign different degrees of importance to positions and to items. Since these distances are easy to aggregate (in our terminology, distances that *characterize* a scoring rule), clustering can be computed very efficiently.

3 Distance and Aggregation

There are a number of ways that can be used to aggregate several rankings into a single one. Some aggregation rules are devised from social choice: the Condorcet method, sorting the items by their Borda score or a generic scoring rule. There are as well many commonly used distance measures for rankings, such as Kendall-tau, foot rule or Spearman.

From a theoretical point of view, the interest is to study if, for a given common aggregation rule (such as plurality), there is a distance measure that it is (implicitly) minimized. In this paper we establish a connection between scoring rules (often used in social choice) and their associated distance measures.

3.1 Aggregation methods

In general terms an aggregation rule is a mapping $g(\sigma_1, ..., \sigma_m)$ from a set of input rankings $\sigma_1, ..., \sigma_m$ to a single "best" ranking summarizing the whole population. As there might be ties in the underlying computations, we allow g to return more than one ranking, so technically g outputs a set of rankings, to be considered equally good.

Many ways of aggregating rankings arise from the field of social choice, where one needs to make a decision for a group of people, aggregating several (usually different) preferences, expressed as a vote in a ballot. Here we focus on rank aggregation using scoring rules.

A scoring rule associates each position $r \in \{1, ..., n\}$ with a score $w(r)$. Items are evaluated by summing up the score they are awarded in each ranking $v(i) = \sum_{j=1}^{m} w(\pi_j(i))$. In order to form an aggregate ranking, items are sorted according to their total score $v(i)$: the ranking π^*_{SR} obtained by a scoring rule is such that $\pi^*_{SR}(i) < \pi^*_{SR}(j)$ iff $v(i) \geq v(j)$ (when ties exist in the overall score, a tie-breaking rule needs to be used).

Borda count (or Borda rule) is a particular type of scoring rule considering weights defined as $w(r) = n - r + 1$ (the item ranked first gets a score of n points, an item in the second position gets $n-1$, and so on). We denote with π^*_{Borda} the ranking obtained by following Borda rule. Borda weights are such that Borda counts for element i are $v(i) = \sum_{u=1}^{m} n - \sigma_u(i) + 1 = m(n+1) - \sum_{u=1}^{m} \sigma_u(i)$. The optimal ranking π^*_{Borda} according to Borda count is such that i precedes j in π^*_{Borda}, formally expressed as $\pi^*_{Borda}(i) < \pi^*_{Borda}(j)$, iff $v(i) \geq v(j)$. It is immediate to show that this is equivalent to $\sum_{u=1}^{m} \sigma_u(i) \leq \sum_{u=1}^{m} \sigma_u(j)$ (an item i is ranked higher than another item j if the overall sum of its positions in the input rankings is less than that of j).

Plurality (sorting items by the number of times that they are ranked first) can be represented as a scoring rule with weights $(1, 0, ..., 0)$; *veto*, sorting items in decreasing order with respect to the number of times they are ranked in the last position, is represented by weights $(1, ..., 1, 0)$, and top-k, that can be modelled as a scoring rule with a weight of 1 in the first k positions and then 0.

We propose a new aggregation method, that we call *Biased Borda* count, parametrized by $z_1, .., z_n$, where one item receives a contribution $n - z_i \sigma_u(i) + 1$ to its score for each ranking σ_u:

$$v_{BB}(i) = m(n+1) - z_i \sum_{u=1}^{m} \sigma_u(i). \tag{2}$$

This allows to "tweak" Borda in order to give some advantage to some items, or penalize others. Obviously when $z_i = 1$ for all i, biased Borda coincides with Borda. The optimal ranking π^*_{BB} with respect to the biased Borda count is such that i precedes j, $\pi^*(i) < \pi^*(j)$, iff $v(i) > v(j)$, or equivalently $z_i \sum_{u=1}^{m} (n - \sigma_u(i) + 1) > z_j \sum_{u=1}^{m} (n - \sigma_u(j) + 1)$, thus if

$$z_i \sum_{u=1}^{m} \sigma_u(i) < z_j \sum_{u=1}^{m} \sigma_u(j). \tag{3}$$

Its interpretation is that an item i is ranked than another item j if the overall sum of its positions in the input rankings, weighted by z_i, is less than that of j, weighted by z_j.

A similar extension can be made considering a scoring rule, instead of Borda, obtaining a *biased scoring rule*.

3.2 Distance Measures for Rankings

Distance measures characterize how different two rankings are; different distances might pose more strength on specific aspects: penalizing the displacements in different ways. For a given distance measure d the total distance from a given ranking π to the a set of rankings $\sigma_1, ..., \sigma_m$ is $\mathrm{D}(\pi; \sigma_1, ..., \sigma_m) = \sum_{u=1}^{m} d(\pi, \sigma_u)$.

A distance function between rankings naturally leads to a way to generating an aggregate ranking; the ranking minimizing this score is chosen as aggregated ranking for the population: $\pi^* = \arg\min_\pi \mathrm{D}(\pi, \sigma_1, ..., \sigma_m)$.

We recall hereafter the usual definition of metric and common generalizations relaxing some of the properties. Note that, while the term distance is often used as a synonym for metric, in the following, we use the former to loosely mean any function that quantifies the difference between elements (rankings in our case), and we explicitly state which distances are metric.

Definition 1. *A function $d : X \times X \to \mathbb{R}$ is a metric on X iff it satisfies the following properties:*

- $d(x, y) \geq 0$ *(non negativity),*
- $d(x, y) = 0$ *iff* $x = y$ *(identity of the indiscernibles),*
- $d(x, y) = d(y, x)$ *(symmetry) and*
- $d(x, y) + d(y, z) \geq d(x, z)$ *(triangular inequality).*

Moreover we have the following relaxations:

- *A* pseudometric d *relaxes the identity of the indiscernibles ($d(x, x) = 0$ but it may holds that $d(x, y) = 0$ for $y \neq x$);*
- *A* quasimetric *relaxes symmetry;*
- *A* semimetric *relaxes the triangular inequality;*
- *A function satisfying non-negativity and $d(x, x) = 0$ is a* premetric.

We are interested in distance measures on rankings; ranking distances d are defined on the S_n (the set of permutation of n elements). Common distance measures for rankings are Kendall tau, footrule[4] and Spearman, that we now recall. We then introduce new distance measures that can express richer notions of similarity/diversity, as giving more weights to the first positions or to the last.

Kendall-tau counts the number of disagreements in terms or pairs between two rankings.

$$d_{KT}(\pi, \sigma) = |\{(i, j) : i > j, (\pi(i) - \pi(j))(\sigma(i) - \sigma(j)) < 0\}| \quad (4)$$

Kendall tau can also be defined as the minimum number of pairwise adjacent transpositions required to bring a ranking into another one [3]. The ranking minimizing the Kendall-tau distance with a set of other rankings, is called *Kemeny* ranking; $\pi^*_{KT} = \arg\min_\pi \sum_{u=1}^m d_{KT}(\pi, \sigma_u)$. Kendall-tau is connected to the Condorcet property: if a Condorcert winner (the item that is pairwise preferred to all other items by the majority of users) exists, minimization of Kendall tau returns such ranking.

Another well known distance measure is *footrule*. Given two rankings π and σ, the displacement for element i is the quantity $|\pi(i) - \sigma(i)|$. The footrule distance measures the total displacement of all elements, computed as

$$d_F(\pi, \sigma) = \sum_{j=1}^{n} |\pi(i) - \sigma(i)|. \quad (5)$$

In this paper we focus on *Spearman distance*, because as it will discussed below in Section 4, it is connected to the aggregation using Borda count. The Spearman distance is defined as taking the squares of the differences:

$$d_S(\pi, \sigma) = \sum_{j=1}^{n} [\pi(j) - \sigma(j)]^2. \quad (6)$$

An interesting observation, that we will use several times in our proofs, is that Spearman can be expressed as follows:

$$d_S(\pi, \sigma) = \frac{n(n+1)(2n+1)}{3} - 2\sum_{i=1}^{n} \pi(i)\sigma(i).$$

In the literature, Sperman distance is often used to measure the correlation between two rankings. Spearman's rank correlation ρ_S is defined as $\rho_S = 1 - \frac{6 d_S(\pi_1, \pi_2)}{n(n^2-1)}$, so that it lies between -1 and 1.

The traditional definition of Spearman distance treats all positions in the same way. Following [4, 9], in order to allow to put more emphasis on some ranks we define a generalization of Spearman distance, that we call *positional Spearman*, giving different weights to rank positions, computed as

$$d_{PS}(\pi, \sigma) = \sum_{i=1}^{n} [w(\pi(i)) - w(\sigma(i))]^2 \quad (7)$$

parametrized by a vector w.

It is easy to verify that footrule and Kendall-tau distance are metrics. Spearman distance, however, does not satisfy the triangular inequality; therefore it is a semimetric. It can be turned into a metric if we take the root of sum of the squares of the distances between positions. We observe that this can also be stated for the proposed positional Spearman ranking distance.

Observation 2. *The positional ranking distance d_{PS} is a semimetric.*

4 Connection between Aggregation Methods and Distance Minimization

Table 1. Distance measures and associated scoring functions.

Aggregation method	Distance measure	Properties
Plurality	d_{PL}	premetric
Top-k	d_{TK}	premetric
Veto	d_V	premetric
Borda	Spearman	semimetric
Scoring rule (distinct weights)	positional Spearman	semimetric
biased Borda	item-weighting Spearman	non negativity, symmetry

The general aggregation problem is that of finding the ranking (permutation of items) that minimizes a given distance measure with respect to several other rankings $\sigma_1, ..., \sigma_m$ given as input.

$$\pi^* = \arg\min_\pi D(\pi; \sigma_1, ..., \sigma_m) = \arg\min_\pi \sum_{j=1}^{m} d(\pi, \sigma_j) \quad (8)$$

Definition 2. *A distance function $d(\pi, \sigma)$ on rankings characterizes a ranking aggregation $g(\sigma_1, ..., \sigma_m)$ iff it holds*

$$\arg\min_{\pi \in S_n} D(\pi; \sigma_1, ..., \sigma_m) = g(\sigma_1, ..., \sigma_m).$$

with $D(\pi; \sigma_1, ..., \sigma_m) = \sum_{u=1}^m d(\pi, \sigma_u)$.

[4] Also known as Spearman's foortule.

In case the aggregation function returns multiple rankings (for example, due to ties in the score obtained with Borda count), then all such rankings should achieve the same minimal value $D^* = \min_\pi D(\pi;\sigma_1,...,\sigma_m)$, and, conversely, if there are several rankings associated with minimum sum-of-distances D^*, these must be returned by g.

We now establish theoretical connections between distance minimization and some well known aggregation methods. In particular, we establish connections between the newly proposed positional Spearman distance (see above in Section 3.2) and aggregation using scoring rules, and between a distance giving different weights to items and aggregation using biased Borda (defined above in Section 3.1).

First of all, it is interesting to note that rank aggregation with Borda count and minimization of Spearman produce the same aggregated ranking.

Observation 3. *[11] The Spearman distance characterizes the Borda rule: $\pi^*_{Borda} = \arg\min_{\pi\in S_n}\sum_{u=1}^{m} d_S(\pi,\sigma_u)$.* [5]

We have noted before that Spearman distance, as defined in Equation 6, is not a metric, as triangular inequalities does not hold. One might wonder if it is possible to "tweak" the Spearman distance to find a metric (satisfying the triangular inequality) with this property. We prove, however, that this is not possible.

Observation 4. *There is no metric characterizing Borda rule.*

We now extend this result to scoring rules assigning arbitrary weights to positions. We derive a novel connection between scoring rules and our proposed positional Spearman distance. Note, however, that the scoring vector must be an injective function (also called one-to-one) in order for the result to hold.

Observation 5. *Assume a scoring rule such that all weights are different; $w(r) \neq w(s)$ if $r \neq s$ with $r, s \in \{1,...,n\}$. The positional Spearman distance with weights w characterizes the scoring rule with the same weights; $\pi^*_{SR} = \arg\min_{\pi\in S_n}\sum_{u=1}^{m} d_{PS}(\pi,\sigma_u)$.*

The previous observations only holds for scoring rules with distinct weights. Why identical weights might be problematic can be seen with an example: consider, for instance, $w = (3,2,2,1)$ where the the second and third position are associated with the same weight; two rankings that differs only by the fact that items on the second and third position are inverted are associated with null distance: $d_{PS}(\langle 1,2,3,4\rangle,\langle 1,3,2,4\rangle) = 0$.

This means that the association fails, notably, for plurality, veto and top-k, that can be represented, respectively, as scoring rules with weights $(1,0,...,0)$, $(0,...,0,1)$ and with a weight vector with 1 in the first k positions and then 0s everywhere. We therefore look for an alternative characterization for these rules, in order to define some distance measures that they implicitly minimize.

Using plurality in our framework for clustering basically means to put together rankings based on the first preferred item. If the number of clusters is lower than the number of different items placed first in any ranking, aggregation will be made by ordering items according to the number of "votes" (number of rankings placing an item first); when assigning rankings to clusters, a ranking with item i in the first position will be assigned to the cluster whose centroid put item i in the highest position.

The following premetric d_{PL} captures this behavior

$$d_{PL}(\pi,\sigma) = \pi(\sigma^{-1}(1)) - 1. \tag{9}$$

Note that d_{PL} is not symmetric. Furthemore, for a given π there are many π' such that $d(\pi,\pi') = 0$; in fact, any σ such that $\sigma^{-1}(1) = \pi^{-1}(1)$: for example $d_{PL}(\langle 1,2,3\rangle,\langle 1,3,2\rangle) = 0$. Therefore d_{PL} is neither a metric nor a semimetric. It holds $d_{PL}(\pi,\pi) = 0$ for any π and $d_{PL}(\pi,\sigma) \geq 0$ (for any π, σ), but the triangular inequality does not hold; thus d_{PL} is a premetric.

Given a set of input rankings $\pi_1,..,\pi_m$, the sum of the distances to a centroid π is $\sum_{t=1}^{m} d_{PL}(\pi,\pi_t) = \sum_{t=1}^{m} \pi(\pi_t^{-1}(1)) - 1$ (for each input ranking π_t, we consider its best ranked element $\pi_t^{-1}(1)$ and look for its position according to π).

We can prove that the ranking obtained by aggregating ranking $\sigma_1,...,\sigma_m$ using plurality is the one that minimizes the sum of distances $D_{PL}(\pi;\sigma_1,...,\sigma_m) = \sum_{t=1}^{m} d_{PL}(\pi,\sigma_t)$.

Observation 6. *The distance d_{PL} characterizes plurality as a method for aggregation of rankings.*

One can wonder if there is another distance that can characterize plurality, with additional properties such as symmetry. In fact, we show that this is not possible.

Observation 7. *There is no semimetric and no quasi metric (hence there is no metric) characterizing plurality.*

For veto, we can define a premetric, analogous to the one we defined for plurality, but that looks for the position of the lowest ranked items.

Observation 8. *The pseudo-distance d_{VT} characterizes the veto rule.*

$$d_{VT}(\pi,\sigma) = n - \pi(\sigma^{-1}(n)). \tag{10}$$

We now consider aggregation with respect to the *top-k* elements.

Observation 9. *The following premetric characterizes the top-k aggregation rule.*

$$d_{topk}(\pi,\sigma) = \sum_{r=1}^{k} \pi(\sigma^{-1}(r)) - \frac{n(n+1)}{2} \tag{11}$$

The constant addend $-\frac{n(n+1)}{2}$ is used in order to satisfy $d(\pi,\pi) = 0$, i.e. to obtain a premetric. Note that d_{topk} is the same as d_{PL} when$= 1$. From the fact that top-k aggregation subsumes plurality, and from Observation 7, it immediately follows:

Observation 10. *There is no semimetric and no quasi metric (hence there is no metric) characterizing top-k.*

In order to characterize biased Borda, we introduce another kind of generalization of Spearman, allowing to give different weights z_i to different items. *Item-weighting Spearman* is defined as:

$$d_{IS}(\pi,\sigma) = \sum_{i=1}^{n} \pi(i)^2 + \sigma(i)^2 - 2z_i\pi(i)\sigma(i) = C_n - 2\sum_{i=1}^{n} z_i\pi(i)\sigma(i) \tag{12}$$

where $C_n = \frac{n(n+1)(2n+1)}{3}$ is regarded as a constant[6], as it depends only on n. The role of the z_i is to tune the impact of position differences, weighting more items that are deemed important. Note that $d_{IS}(\pi,\sigma) = d_S(\pi,\sigma)$ if all weights z_i are set to 1.

We now establish the connection between this new distance measure and the biased Borda aggregation rule presented before.

[5] Proofs are provided in the appendix.

[6] In fact another constant might be used, we use C_n in order to yield Spearman distance values if the z_i are set to 1.

Observation 11. *Item-weighted Spearman characterizes the biased Borda rule.*

Note that d_{IS} is not even a premetric, as $d_{IS}(\pi, \pi)$ can (and will often) yield a value different than zero. d_{IS} is symmetric and non-negative. A much "nicer" distance function is the following

$$\hat{d}_{IS}(\pi, \sigma) = \sum_{i=1}^{n} z_i [\pi(i) - \sigma(i)]^2$$

that is a semimetric (notice that $\hat{d}_{IS}(\pi, \sigma) = d_{IS}(\pi, \sigma) - d_{IS}(\pi, \pi) - d_{IS}(\sigma, \sigma)$); however, we could not find a characterization for this distance, and its optimization seems to be rather hard.

The theoretical results are summarized in Table 1.

5 Application to Clustering

Table 2. Computation times for distance-based clustering (randomly generated rankings; 10 runs).

		$k = 5$ (number of clusters)			
m	n	aggregator	distance	time	iterations
100	10	plurality	d_{PL}	0.01	2.3
		Borda	Spearman	0.35	9.3
		scoring rule	positional Spearman	0.31	7.8
		biased Borda	item-w Spearman	0.48	8.7
100	20	plurality	d_{PL}	0.02	2.8
		Borda	Spearman	0.43	6.7
		scoring rule	positional Spearman	0.45	6.4
		biased Borda	item-w Spearman	0.81	8.0
2000	10	plurality	d_{PL}	0.16	2.5
		Borda	Spearman	3.96	5.7
		scoring rule	positional Spearman	7.10	8.4
		biased Borda	item-w Spearman	9.41	7.7
2000	20	plurality	d_{PL}	0.17	2.4
		Borda	Spearman	36.55	28.5
		scoring rule	positional Spearman	17.85	13.0
		biased Borda	item-w Spearman	28.06	14.1
		$k = 10$ (number of clusters)			
m	n	aggregator	distance	time	iterations
100	10	plurality	d_{PL}	0.02	2.5
		Borda	Spearman	0.60	8.6
		scoring rule	positional Spearman	0.62	8.0
		biased Borda	item-w Spearman	0.75	6.8
100	20	plurality	d_{PL}	0.03	2.7
		Borda	Spearman	0.96	7.3
		scoring rule	positional Spearman	1.09	7.2
		biased Borda	item-w Spearman	1.65	7.6
2000	10	plurality	d_{PL}	0.40	2.6
		Borda	Spearman	12.19	7.6
		scoring rule	positional Spearman	17.20	10.7
		biased Borda	item-w Spearman	20.07	9.2
2000	20	plurality	d_{PL}	0.34	2.7
		Borda	Spearman	93.53	35.0
		scoring rule	positional Spearman	58.49	19.7
		biased Borda	item-w Spearman	62.78	15.0

The theoretical connection between aggregation rules and distance minimization (Section 4) can be used to perform clustering in an efficient way. Given the theoretical observations provided in the previous section, rank aggregation minimizing Spearman and its proposed generalization (positional Spearman and item-weighting Spearman) consist in a scoring rule (Borda count in the case of classic Spearman).

With plurality, veto and top-k, from a practical point of view, the aggregation is very easy and intuitive: for plurality, we sort the items according to the number of times they are top-ranked. Similarly, for veto we rank items according to the number of times they are not last in a user ranking, and similarly for top-k.

This contrasts with the computational effort required by the minimization of Kemeny or Footrule. The optimization of Footrule distance measure can be formulated as an assignment problem; where the "cost" of placing item i at place r is the total number of displacements compared to the input rankings. Footrule aggregation can therefore be computed rather efficiently with linear programming techniques. Minimization of Kendall tau is NP-hard; see for instance [6] for complexity results A good proxy for Kemeny is the use of footrule: it is known that $d_{KT} \leq d_F \leq 2d_{KT}$ [3]; therefore the minimization of Footrule can provide a ranking that is also good with respect to Kendall.

Table 3. Computation times for distance-based clustering (sushi dataset; 20 runs).

k	aggregator	distance	time	iterations
2	plurality	d_{PL}	0.21	3.00
	Borda	Spearman	6.99	10.10
	scoring rule	positional Spearman	4.71	6.05
	biased Borda	item-w Spearman	5.99	5.55
3	plurality	d_{PL}	0.30	3.00
	Borda	Spearman	13.26	13.15
	scoring rule	positional Spearman	8.58	7.05
	biased Borda	item-w Spearman	13.75	8.20
5	plurality	d_{PL}	0.47	3.00
	Borda	Spearman	19.92	11.70
	scoring rule	positional Spearman	16.08	7.95
	biased Borda	item-w Spearman	30.34	10.85
10	plurality	d_{PL}	0.98	3.00
	Borda	Spearman	49.17	14.15
	scoring rule	positional Spearman	37.55	9.50
	biased Borda	item-w Spearman	71.43	12.95

In the first experiment[7] we consider synthetic ranking data of different sizes, in order to assess the computation time. Clustering with these distance-based techniques is very efficient. Table 2 reports computation times (in seconds) and the number of iterations before convergence (averaged over 10 runs; the weight vector z for Biased Borda and item-weighting Spearman is randomly sampled at each run) for different combinations of aggregators/distances, different number of clusters and different values of m and n. Interestingly convergence seems to be slower (leading to higher computation times) for clustering based on item-weighting Spearman (aggregation with biased Borda count), but not always. Table 3 reports similar statistics for the sushi dataset.

In the second set of experiments, we consider real data. In the *sushi dataset*[8] 5000 users have been asked to rank a sets of items (sushis). [8] from the most to least preferred.

First of all we consider the median aggregate ranking considering the whole user population. Using Borda rule, the central ranking among the overall population is the following

$$\langle 8,3,1,6,2,5,9,4,7,10 \rangle$$

(Sushi n.8 is the most preferred, followed by sushi n.3, then sushi n.1, ...). Instead, using plurality, we derive the following ranking

$$\langle 8,5,2,6,1,3,4,7,9,10 \rangle$$

where sushi 8 is again ranked first, followed this time by sushi n. 5 and 2 that gain several positions (compared to using Borda), as they are often ranked first by several users (sushi n. 5 is ranked first by 747 users; sushi n. 2 by 550), at the expense of sushi 1 and 3 (ranked first by 458 and 404 respectively). We then perform rank aggregation using a scoring rule with the following weights $w_1 = (20,15,10,7,5,4,3,2,1,0)$. This weight vector is a convex sequence,

[7] All experiments are programmed in MATLAB and executed on a MacBook Pro (late 2013 version) with processor Intel Core i7 with 8 GB of memory.

[8] Available at http://www.kamishima.net/sushi/.

satisfying $w_i - w_{i-1} \geq w_{i+1} - w_i$ for all positions $i = 2, ..., n-1$, meaning that the difference between adjacent positions towards the bottom is lower than difference towards the top). With this weight we obtain the following aggregate ranking.

$$\langle 8,3,6,5,1,2,4,9,7,10 \rangle$$

We investigate which clusters are produced with distance-based clustering with different distances. Since our distance-based clustering approach returns a local optimum, we repeat the algorithm a number of times (typically 10 or 20) and store the clustering assignment associated with the lowest sum of distances to the closest centroid. We now want to show that the clustering methods proposed in this work allow greater flexibility (than currently used methods) as, by using our propose distance measures, it is possible to represent the higher importance of some items or to some positions; the resulting clusters will then display the desired characteristics.

We obtain the following centroids with Borda:

$$\langle 8, 3, 1, 9, 4, 6, 7, 10, 5, 2 \rangle,$$
$$\langle 5, 8, 6, 3, 2, 1, 4, 9, 7, 10 \rangle,$$
$$\langle 1, 7, 2, 4, 10, 9, 3, 6, 8, 5 \rangle,$$
$$\langle 8, 2, 3, 1, 9, 4, 6, 7, 5, 10 \rangle$$

associated to clusters of size 1110, 2214, 500 and 1176.

We obtain the following centroids when clustering with the scoring rule associated to positional Spearman with weights w_1

$$\langle 6, 8, 1, 3, 4, 9, 2, 7, 5, 10 \rangle,$$
$$\langle 5, 8, 6, 1, 2, 3, 4, 9, 7, 10 \rangle,$$
$$\langle 8, 3, 1, 9, 4, 7, 2, 6, 5, 10 \rangle,$$
$$\langle 2, 8, 1, 7, 3, 4, 9, 6, 5, 10 \rangle$$

Clusters of size 1658, 1021, 855 and 1466, respectively. The second cluster is very similar to the second cluster retrieved by Borda (apart from positions of sushi $1, 2, 3$), but its size (i.e. number of sushi assigned to this cluster) is about an half. Compared to Borda, the new centroids display more agreement at the bottom (sushi n.10 is always ranked last), but at the same time there is a greater variability in the first positions (the top element in each centroid is different).

Table 4. Difference between clusterings measured by the Rand index obtained with different aggregators/distances (sushi dataset; $k = 5$).

	Pl	Borda	s. rule w_1	s. rule w_2	s. rule w_3
Plurality	–	0.46	0.50	0.47	0.41
Borda	–	–	0.27	0.29	0.33
Scoring rule w_1	–	–	–	0.12	0.21
Scoring rule w_2	–	–	–	–	0.14
Scoring rule w_3	–	–	–	–	–

We now analyze how different weight parameters induce different clusterings. In Table 4 we compute the Rand index (measuring the fraction of pair of rankings whose assignment agrees in the two clustering) between each pair of clusterings obtained[9] with plurality, Borda, and scoring distances with the following weight vectors

$$w_1 = (20, 15, 10, 7, 5, 4, 3, 2, 1, 0),$$
$$w_2 = (20, 12, 8, 6, 5, 4, 3, 2, 1, 0),$$
$$w_3 = (512, 256, 128, 64, 32, 16, 8, 4, 2, 1).$$

all with the associated distance measures.

6 Discussion

Clustering rank data can be seen as unsupervised preference learning. A key element is rank aggregation, the problem of combining the ranked preferences of different experts or users into a single 'consensus' ranking; it can be thought of as the unsupervised analogue to regression. This paper deals with clustering methods that iteratively use aggregation to compute a centroid for each cluster, and then assign rankings to the cluster whose centroid is closest (according to some distance measure between rankings). The question about the connection between common aggregation methods for rankings (such as scoring rules) and distance measures naturally arises.

In this paper we provided a taxonomy of distance measures to be used for clustering preference rankings, that are associated with scoring rules as aggregation method. We extended the result about the connection between Borda rule and minimization of Spearman distances to scoring rules and a new measure that give weights to positions. We consider the case of plurality, veto and top-k. We also introduced a new aggregation rule, biased Borda, giving more advantage to specific items, and show how it can be characterized.

The clustering methods proposed in this work allow greater flexibility (than currently used methods) as, by using our proposed distance measures, it is possible to represent the higher importance of some items or to some positions; the resulting clusters will then display the desired characteristics.

In future works, we plan to extend the empirical analysis of clustering with real data; we will also consider methods to deal with partial rankings (rankings defined on a subset of items): this is crucial in order to deal with real applications.

Moreover we are planning to consider interactive elicitation of distance weights, posing questions of the kind *"Are rankings π_1 and π_2 more similar than rankings σ_1 and σ_2?"* to the user in order to learn the distance function interactively. This idea is briefly mentioned next.

6.1 Handling Parameter Uncertainty

The proposed new measures for representing distances between rankings are more expressive (than traditional distance measures) as they can model different degrees of importance associated to ranking positions. However, in practical situations it might not be so obvious how to assign the parameters employed by positional Spearman and item-weighting Spearman. We are currently considering situations in which limited information is known about the weights, but we still may need to perform clustering.

The idea is to reason about the set of feasible parameters given the current information, represented by constraints, and cast the setting into a robust optimization problem (in similar ways as in utility-based recommender systems [13]). For positional Spearman, it is natural to state that weights are decreasing with rank positions: $w(1) \geq w(2) \geq ... \geq w(n)$. Another reasonable constraint[10] is to require $w(1), ..., w(n)$ to constitute a convex sequence: $w(r) \leq \frac{w(r-1)+w(r+1)}{2}$ for each position $r = 2,..,n-1$. Additional information might be provided by a user expressing that two rankings π_1, π_2 are more similar to each other than another pair of rankings σ_1, σ_2; the semantics is that the distance between π_1 and π_2 is lower than between σ_1 and σ_2. This information can be encoded by constraints on the feasible parameters. With positional Spearman this gives quadratic constraints on the feasible weights $w(1), ..., w(n)$:

[9] In each run of the clustering algorithm, computations are repeated 10 times and the best local minimum is picked.

[10] This is the case, for instance, of the scoring rules used in common racing competitions.

$$d_{PS}(\pi_1,\pi_2) \leq d_{PS}(\sigma_1,\sigma_2) \leftrightarrow$$
$$\sum_{i=1}^{n} w(\pi_1(i))w(\pi_2(i)) \geq \sum_{i=1}^{n} w(\sigma_1(i))\, w(\sigma_2(i)).$$

In the case of item-weighting, the information about a pair of rankings being more similar than another pair, $d_{IS}(\pi_1,\pi_2) \leq d_{IS}(\sigma_1,\sigma_2)$, is encoded linear constraints:

$$\sum_{i=1}^{n} z_i \pi_1(i)\pi_2(i) \geq \sum_{i=1}^{n} z_j \sigma_1(i)\sigma_2(i).$$

We plan to investigate strategies to generate clusters of rank data without precise weight information, where only a number of statements comparing the degree of similarity between pairs is given, handling such constraints.

6.2 Related Works

Axiomatic treatment of the median ranking from a point of view of social choice is given in [1]. While our work focuses on non-parametric distance models, other common approach relies on probabilistic models [2], including Babington-Smith and Mallows models [10].

The idea of looking aggregation techniques in term of minimization of distance measure is known as *distance rationalizability* in social choice [5]. The difference is that in our clustering application, we are interested in an aggregation that produces a ranking, while in social choice most of the emphasis is on the winner, the elected candidate.

Other works also considered extending common distances in some ways. In [12] methods of ranking aggregation are extended in order to exploit similarity information between ranked items. Local Kemenization [4] computes a locally optimal ranking where swapping two adjacent items cannot further reduce Kendall tau distance.

Kamishima and Akaho [7] provide some efficient strategies for clustering rankings, also accounting for partial rankings. The generalized distance functions presented in [9] are a rich generalization of Footrule and Kendall whose expressivity is similar to the distances proposed here (moreover, they can define a specific weight for swapping two particular items). Here, we focus on Spearman distance and its generalization; this turns out to be advantageous because of the connection with scoring rules.

REFERENCES

[1] Jean-Pierre Barthélemy and Bernard Monjardet. The median procedure in cluster analysis and social choice theory. *Mathematical Social Sciences*, 1(3):235–267, 1981.

[2] Douglas E Critchlow, Michael A Fligner, and Joseph S Verducci. Probability models on rankings. *Journal of Mathematical Psychology*, 35(3):294–318, 1991.

[3] Persi Diaconis and R. L. Graham. Spearman's Footrule as a Measure of Disarray. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(2), 1977.

[4] Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. In Vincent Y. Shen, Nobuo Saito, Michael R. Lyu, and Mary Ellen Zurko, editors, *WWW*, pages 613–622. ACM, 2001.

[5] Edith Elkind, Piotr Faliszewski, and Arkadii M. Slinko. On distance rationalizability of some voting rules. In Aviad Heifetz, editor, *TARK*, pages 108–117, 2009.

[6] Olivier Hudry. Complexity of computing median linear orders and variants. *Electronic Notes in Discrete Mathematics*, 42:57–64, 2013.

[7] Toshihiro Kamishima and Shotaro Akaho. Efficient clustering for orders. In Djamel A. Zighed, Shusaku Tsumoto, Zbigniew W. Ras, and Hakim Hacid, editors, *Mining Complex Data*, volume 165 of *Studies in Computational Intelligence*, pages 261–279. Springer, 2009.

[8] Toshihiro Kamishima, Hideto Kazawa, and Shotaro Akaho. Supervised ordering - an empirical survey. In *ICDM*, pages 673–676. IEEE Computer Society, 2005.

[9] Ravi Kumar and Sergei Vassilvitskii. Generalized distances between rankings. In Michael Rappa, Paul Jones, Juliana Freire, and Soumen Chakrabarti, editors, *WWW*, pages 571–580. ACM, 2010.

[10] Tyler Lu and Craig Boutilier. Learning Mallows Models with Pairwise Preferences. In Lise Getoor and Tobias Scheffer, editors, *ICML*, pages 145–152. Omnipress, 2011.

[11] John I Marden. *Analyzing and modeling rank data*. CRC Press, 1996.

[12] D. Sculley. Rank Aggregation for Similar Items. In *SDM*. SIAM, 2007.

[13] Paolo Viappiani and Craig Boutilier. Regret-based optimal recommendation sets in conversational recommender systems. In *Proceedings of the 2009 ACM Conference on Recommender Systems, RecSys 2009, New York, NY, USA, October 23-25, 2009*, pages 101–108, 2009.

A Proofs

The following observation is used a number of times.

Observation 12. *Given a set of* distinct *numbers $x_1,..x_n$ and a set $y_1,..,y_n$ ordered such that $y_1 \leq y_2 \leq ... \leq y_n$ (with possible ties), the permutation π of x* maximizing $\sum_{i=1}^{n} \pi^{-1}(i)\, y_i = \sum_{i=1}^{n} x_i y_{\pi(i)}$ *is the permutation $\langle x_{(1)},...,x_{(n)}\rangle$ such that $x_{(1)} < x_{(2)} < ... < x_{(n)}$ (the permutation sorting the elements of x in increasing order).*

The permutation π of x minimizing *the some formula, is $\langle x_{(n)},...,x_{(1)}\rangle$ (the permutation sorting the elements of x in decreasing order).*

We now provide proofs for the properties stated in the paper.

Proof that Spearman distance characterizes Borda count. Borda weights $w_i = n-i+1$ are such that Borda counts for element i are $v(i) = \sum_{u=1}^{m} n-\sigma_j(i)+1 = n(m+1)-\sum_{j=1}^{m}\sigma_j(i)$. The optimal ranking π^* according to Borda count is such that i precedes j in π^* ($\pi^*(i) < \pi^*(j)$ iff $v(i) \geq v(j)$, or equivalently iff $\sum_{u=1}^{m}\sigma_u(i) \leq \sum_{u=1}^{m}\sigma_u(i)$.

We are interested in the ranking π^* that minimizes the sum of the Spearman distance with a number of input ranking $\sigma_1,..,\sigma_m$: $\pi^* = \arg\min_\pi \sum_{u=1}^{m} d_S(\pi,\sigma_u)$.

Let $C_n = \sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$. The Spearman distance between two rankings π and σ can be rewritten as follows

$$d_S(\pi,\sigma) = \sum_{i=1}^{n}[\pi(i)-\sigma(i)]^2 = \sum_{i=1}^{n}\pi(i)^2+\sigma(i)^2-2\pi(i)\sigma(i) =$$
$$= \frac{n(n+1)(2n+1)}{3} - 2\sum_{i=1}^{n}\pi(i)\sigma(i) = 2\Big(C_n - \sum_{i=1}^{n}\pi(i)\sigma(i)\Big).$$

The sum of the Spearman distances between π and $\sigma_1,..,\sigma_m$ is then

$$D_S(\pi;\sigma_1,...,\sigma_m) = \sum_{u=1}^{m} d_S(\pi,\sigma_u) = 2\Big(m\, C_n - \sum_{u=1}^{m}\sum_{i=1}^{n}\pi(i)\sigma_u(i)\Big). \tag{13}$$

Therefore the ranking with minimum total Spearman distance with respect to a set of rankings $\sigma_1,..,\sigma_m$ is

$$\arg\min_\pi D_S(\pi;\sigma_1,...,\sigma_m) = \arg\max_\pi \sum_{i=1}^{n}\sum_{u=1}^{m}\pi(i)\sigma_u(i) =$$
$$= \arg\max_\pi \sum_{i=1}^{n}\pi(i)\sum_{u=1}^{m}\sigma_u(i).$$

The result follows by applying Observation 12 to the last expression; the aggregated ranking π will be such that i precedes j, $\pi(i) < \pi(j)$, if $\sum_{u=1}^{m}\sigma_u(i) \leq \sum_{u=1}^{m}\sigma_u(i)$, thus this is the same ranking as aggregation using Borda count.

Proof that there is no metric that characterizes Borda count. Let d be a semimetric (satisfying non negativity, identity of the indiscernibles and symmetry) between rankings. We prove that if d characterizes Borda rule, then it cannot satisfy the triangular inequality. Consider the following population of rankings: $\sigma_1 = \langle 1, 2, 3\rangle$ and $\sigma_2 = \langle 3, 1, 2\rangle$. Application of Borda gives the following scores to items: $v(1) = 5, v(2) = 3, v(3) = 4$ yielding the optimal ranking $\pi^* = \langle 1, 3, 2\rangle$.

If d characterizes Borda rule, then it must hold that $D(\pi^*; \sigma_1, \sigma_2) < D(\pi; \sigma_1, \sigma_2)$, for any $\pi \in S_n, \pi \neq \pi^*$. In particular, π^* must compare favorably with respect to σ_1: $\sum_{u=1,2} d(\pi^*, \sigma_u) < \sum_{u=1,2} d(\sigma_1, \sigma_u)$. Note that since d is a semimetric, we must have $d(\pi^*,\sigma_1) = d(\sigma_1,\pi^*)$ and $d(\sigma_1,\sigma_1) = 0$; it then follows that $d(\sigma_1, \pi^*) + d(\pi^*, \sigma_2) < d(\sigma_1, \sigma_2)$, so the triangular inequality is not satisfied.

Proof that d_{PL} characterizes plurality. Let α_i^1 be the number of input rankings among $\sigma_1, ..., \sigma_m$ for which i is ranked first; $\alpha_i^1 = |\{\sigma : \sigma(i) = 1\}|$. It follows $\sum_{t=1}^{m} d_{PL}(\pi, \sigma_t) = \sum_{t=1}^{m} \pi(\sigma_t^{-1}(1)) - 1 = \sum_{t=1}^{m}\left[\left(\sum_{i=1}^{n}\pi(i)\, I[\sigma_t(i) = 1]\right) - 1\right] = \sum_{i=1}^{n}\pi(i) \sum_{t=1}^{m} I[\sigma_t(i) = 1] - m = \sum_{i=1}^{n}\alpha_i^1\pi(i) - m$ (where I is the indicator function). Therefore we have

$$\pi^* = \arg\min_{\pi \in S_n} \sum_{t=1}^{m} d_{PL}(\sigma_t, \pi) = \arg\min_{\pi \in S_n} \sum_{i=1}^{n} \alpha_i^1 \pi(i).$$

By using Observation 12, it follows that the permutation π^* minimizing the sum of the distances must be such that $\pi(i) < \pi(j)$ if $f_i > f_j$. This means that the item that is ranked first from the highest number of input rankings, will be placed first in the aggregated ranking. The next item in the aggregate ranking will be the one that is ranked first from the second highest number of input rankings, and so on. This is exactly the result of aggregation when using plurality.

Proof that no semimetric characterizes plurality. Consider a population of two rankings $\sigma_1 = \langle 1, 2, 3\rangle$ and $\sigma_2 = \langle 2, 3, 1\rangle$. According to plurality, the best rankings obtained by aggregating σ_1 and σ_2 are σ_1 itself and the ranking $\langle 2, 1, 3\rangle$ (as they both rank items 1 and 2 - each mostly preferred exactly one time in σ_1, σ_2 - before item 3 - that is never maximally preferred). Now, consider a premetric d and assume that it characterizes plurality. Since σ_2 is not a optimal ranking according to plurality, the sum of the distances between σ_1 and the population must be strictly lower than the sum of the distances from σ_2

$$d(\sigma_1, \sigma_1) + d(\sigma_1, \sigma_2) < d(\sigma_2, \sigma_2) + d(\sigma_2, \sigma_1)$$

from which (since $d(\pi, \pi) = 0$, d being a premetric) it follows $d(\sigma_1, \sigma_2) < d(\sigma_2, \sigma_1)$, hence any d characterizing plurality cannot be symmetric.

Proof that d_{VT} characterizes veto The proof is analogous to the case of plurality. Let α_i^n be the number of input rankings (users) $\sigma_1, ..., \sigma_m$ in which i is ranked in the last position. $\sum_{t=1}^{m} d_{VT}(\pi, \sigma_t) = nm - \sum_{t=1}^{m}\pi(\sigma_t^{-1}(n)) = nm - \sum_{t=1}^{m}\sum_{i=1}\pi(i)\, I[\sigma_t(i) = n] = nm - \sum_{i=1}^{n}\alpha_i^n\pi(i)$ (where I is the indicator function). Therefore

$$\arg\min_{\pi \in S_n} \sum_{t=1}^{m} d_{VT}(\sigma_t, \pi) = \arg\max_{\pi \in S_n} \sum_{i=1}^{n} \alpha_i^n \pi(i)$$

and the proof follows by applying Observation 12 (similarly as in the case of d_{PL}); since we are maximizing, the objective is attained according to an increasing order of α_i^n, in accordance with the veto rule.

Proof that d_{topk} characterizes top-k aggregation. The proof analogous to the case of plurality. Let $\alpha_i^{<k}$ be the number of input rankings (users) $\sigma_1, ..., \sigma_m$ in which i is ranked among the top-k $|\{\sigma : \sigma(i) \leq k|\}$. $\sum_{t=1}^{m} d_{topk}(\pi, \sigma_t) = \sum_{t=1}^{m}\sum_{r=1}^{k} \pi(\sigma^{-1}(r)) - 1 = \sum_{i=1}^{n}\pi(i)\sum_{t=1}^{m}\ I[\sigma_t(i) \leq k]\ - mk = \sum_{i=1}^{n}\pi(i)\alpha_i^{<k}\ - mk$. Then we have

$$\arg\min_{\pi \in S_n} \sum_{t=1}^{m} d_{VT}(\sigma_t, \pi) = \arg\min_{\pi \in S_n} \sum_{i=1}^{n} \alpha_i^{<k} \pi(i)$$

and the proof follows by applying Observation 12 as in the case of d_{PL}.

Proof that the positional Spearman distance characterizes scoring rules. The optimal ranking π^* according to a scoring rule with weights w is such that i precedes j in π^* iff $v(i) \geq v(j)$, or equivalently iff $\sum_{u=1}^{m} w(\sigma_u(i)) \geq \sum_{u=1}^{m} w(\sigma_u(j))$.

The positional Spearman distance (taking into account weights associated to positions) between two rankings π and σ can be rewritten as follows

$$d_{PS}(\pi,\sigma) = \sum_{i=1}^{n}[w(\pi(i)) - w(\sigma(i))]^2 = 2Z_n^w - 2\sum_{i=1}^{n} w(\pi(i))\, w(\sigma(i))$$

where we let Z_n^w be $\sum_{i=1}^{n} w(i)^2$; note that, for any permutation π, it holds the $\sum_{i=1}^{n} w(\pi(i))^2 = Z_n^w$, with Z_n^w depending only on the weights w and the number of items n. Proceeding as before in the case of standard Spearman, the ranking with minimum total distance is

$$\arg\min_{\pi} \sum_{u=1}^{m} d_{PS}(\pi, \sigma_u) = \arg\max_{\pi} \sum_{i=1}^{n}\sum_{u=1}^{m} w(\pi(i)) w(\sigma_j(i)) = \tag{14}$$

$$= \arg\max_{\pi} \sum_{i=1}^{n} w(\pi(i)) \sum_{u=1}^{m} w(\sigma_j(i)). \tag{15}$$

The result follows by applying Observation 12 to Equation 15, similarly as before, but this time the assumption that w is injective is crucial for obtaining the result.

Proof that item-weighting Spearman distance characterizes the biased Borda count. The sum of distances is obtained as

$$\sum_{u=1}^{m} d_{IS}(\pi, \sigma_u) = mC_n - 2\sum_{i=1}^{n} \pi(i)\, z_i \sum_{u=1}^{m} \sigma_u(i)$$

therefore

$$\arg\min_{\pi} \sum_{u=1}^{m} d_{IS}(\pi, \sigma_u) = \arg\max_{\pi} \sum_{i=1}^{n} \pi(i)\, z_i \sum_{u=1}^{m} \sigma_u(i).$$

Using the argument used in the previous proofs (Observation 12), the optimal ranking π^* minimizing this sum of distances, is such that i precedes j, $\pi^*(i) < \pi^*(j)$, iff $z_i \sum_{u=1}^{m} \sigma_u(i) < z_j \sum_{u=1}^{m} \sigma_u(j)$, that is exactly what characterizes the biased Borda rule (Equation 3).

An interactive approach for multiple criteria selection problem

Anıl Kaya[1], **Özgür Özpeynirci**[2] and **Selin Özpeynirci**[3]

Abstract. In this study, we develop an interactive algorithm for the multiple criteria selection problem that aims to find the most preferred alternative among a set of known alternatives evaluated on multiple criteria. We assume the decision maker (DM) has a quasi-concave value function that represents his/her preferences. The interactive algorithm selects the pairs of alternatives to be asked to the DM based on the estimated likelihood that an alternative is preferred to another one. After the DM selects the preferred alternative, a convex cone is generated based on this preference information and the alternatives dominated by the cone are eliminated. Then, the algorithm updates the likelihood information for the unselected pairwise questions. We present the algorithm on an illustrative example problem.

1 Introduction

The multiple criteria selection problem aims to identify the most preferred alternative among a set of alternatives that are evaluated on multiple criteria. This problem appears in many real life situations such as selecting the best house to buy for an individual or selecting the best supplier to purchase the raw materials for a company (see for example [2]). We propose an interactive algorithm assuming that the decision maker (DM) has an underlying quasi-concave value function that is not known explicitly. As mentioned in [13], quasi-concave utility functions are more general compared to other utility functions like pseudoconcave, concave or linear and they require less restrictive assumptions regarding the DM's behaviour. Also there are MCDM methods assuming a concave value function (see for example [1]).

The algorithm developed in this study interacts with the DM by asking pairwise comparison questions and eliminates the inferior alternatives using the convex cones generated by the gathered preference information. There are interactive approaches in the literature utilizing the convex cones; including [8],[13], [10], and [12]. Taner and Köksalan [18] conduct detailed experiments on question selection and cone generation procedures.

In this study, we develop an interactive algorithm that picks the alternatives to be used in the pairwise comparisons based on a likelihood approach. For each pair of alternatives and for each possible response of the DM, the algorithm computes (i) the likelihood of this response (ii) the number of alternatives that will be eliminated by this response of DM by solving a number of mathematical programming problems.

The outline of the paper is as follows: in the next section, we review the related literature and provide the necessary background. In Section 3, we discuss the likelihood computations using mathematical programming problems. In Section 4, we present the interactive algorithm. Section 5 presents the application of the interactive algorithm on an illustrative problem and Section 6 concludes the paper.

2 Literature Review and Background

In this section, we present the review of the multiple criteria decision making (MCDM), convex cone method in MCDM and interactive algorithms. MCDM methods aim to solve decision problems involving multiple criteria. Generally, there is not a unique optimal solution for MCDM problems that optimizes all objectives simultaneously. We refer to [17], [16] and [4] for a detailed discussion on MCDM theory, methods and applications.

Suppose there are m decision alternatives evaluated on p criteria. For each criteria, we assume a higher score is better. We also assume there is one decision maker (DM) who owns the problem and answers the pairwise comparison questions.

2.1 Convex Cone Approach

Korhonen, Wallenius and Zionts [12] develop an algorithm that generate cones depending on the responses of the decision maker who has a quasi-concave increasing utility function. They propose to use quasi-concave utility function since it represents the human nature well. Their algorithm generates convex cones and eliminates inferior alternatives based on Theorem 1.

Theorem 1. *(Korhonen, Wallenius and Zionts [12]) Assume a quasi-concave and nondecreasing function f(x) defined in a p-dimensional Euclidean space* $\mathbb{R}^p$*. Consider distinct points* $x_i \in \mathbb{R}^p, i = 1, ..., m$*, and any point* $x^\star \in \mathbb{R}^p$ *and assume that* $f(x_k) < f(x_i), i \neq k$*. Then, if* $\epsilon \geq 0$ *in the following linear programming problem*

$$\begin{aligned} &Max\ \epsilon \\ &\quad s.t. \\ &\qquad \sum_{\substack{i=1 \\ i\neq k}}^{m} \mu_i(x_k - x_i) - \epsilon \geq x^\star - x_k \\ &\qquad\qquad \mu_i \geq 0, \forall i \end{aligned}$$

It follows that $f(x_k) \geq f(x^\star)$*.*

[1] İzmir University of Economics, Department of Logistics Management email: anll.k@hotmail.com

[2] İzmir University of Economics, Department of Logistics Management email: ozgur.ozpeynirci@ieu.edu.tr

[3] İzmir University of Economics, Industrial Engineering Department email: selin.ozpeynirci@ieu.edu.tr

Suppose, in Figure 1, DM prefers x_1 to x_2. We use this preference information to generate a convex cone which corresponds to Region A. Any alternative in this region is dominated by the cone, will be inferior and eliminated.

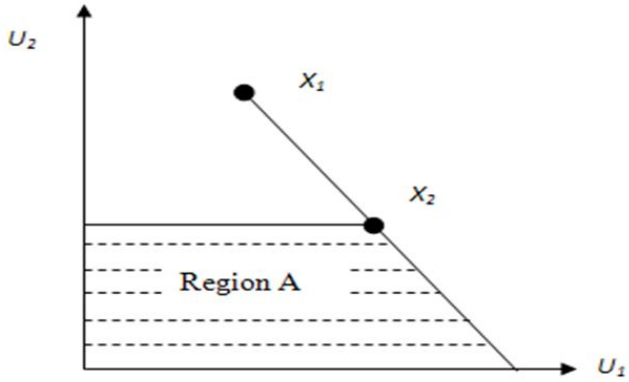


Figure 1. The Illustration of Cone, Source: [12]

The algorithm of Korhonen, Wallenius and Zionts [12] picks a reference alternative and asks DM to compare this alternative with the adjacent alternatives. The algorithm stops if the reference alternative is preferred to all adjacent alternatives, otherwise it moves to another alternative.

Köksalan, Karwan and Zionts [8] construct and use dummy alternatives in order to reduce the number of questions. They combine the approach of Korhonen, Wallenius and Zionts [12] with the idea of using dummy alternatives in cone generators. The proposed dummy alternatives are convex combinations of the existing alternatives. In the method of Köksalan, Karwan and Zionts [8], instead of comparing x_1 and x_2, x_d and x_2 are compared (Figure 2). If DM prefers x_1 to x_2, the alternatives in Region A are eliminated. However, if DM prefers x_d to x_2, the alternatives in Regions A and B are eliminated.

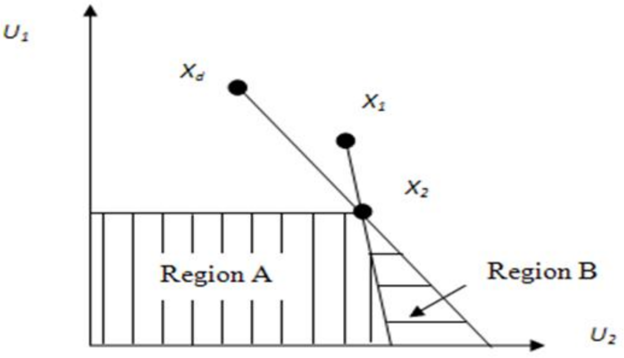


Figure 2. Cones with dummy alternatives, Source: [12]

Köksalan and Taner [10] make improvements to reduce required number of pairwise questions. They develop variations of the dummy alternatives. They use dummy alternatives that are dominated alternatives as cone generator. Instead of comparing x_1 and x_2, x_1 and x_d are compared in Figure 3. If DM prefers x_1 to x_2, alternatives in Region A will be eliminated. If DM prefers x_1 to x_d, alternatives in Regions A and B will be eliminated.

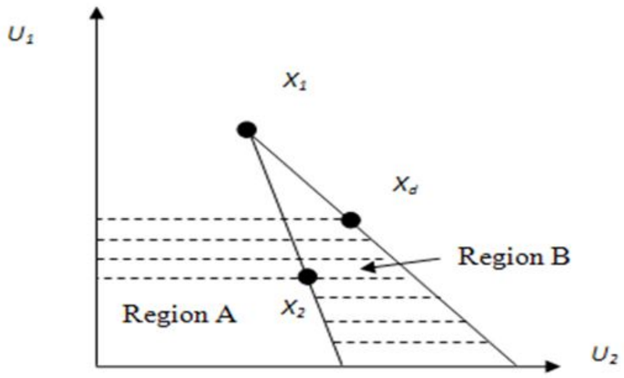


Figure 3. Cones with dummy alternatives, Source: [10]

Köksalan [7] develop an approach to reduce the total number of required questions. The decision maker has a quasiconcave utility function. He uses two different utility functions, one is quadratic and the other is Tchebyshev utility function. He uses the ideal point as an evaluation criterion. The alternatives selected as cone generator are closest to an ideal point in Euclidean distance. In each iteration, he uses alternatives that maximize utility functions, change with the least preferred cone generator.

Taner and Köksalan [18] conduct experiments to see the effect of cones. They use two different utility functions: quadratic and linear utility function. They estimate utility functions using the decision maker's preferences. They select alternatives that have high rankings. Their approach has two variations: finding the best alternative and finding the worst alternative.

Karsu [5] reviews the theory of convex cones approach. She provides a discussion of interactive algorithms utilizing convex cones and discusses further research directions.

2.2 Interactive Algorithms

Interactive algorithms gather information from the DM when needed throughout the algorithm. In the following steps, they use this information to make a decision. The preferences of DM provide information about the value function.

An extensive literature is available on interactive algorithms for multiple criteria sorting problems. Köksalan and Ulu [11] propose an interactive approach, assuming an underlying additive linear utility function for the sorting problem. They use the preferences of the DM to assign alternatives to different categories. Köksalan and Özpeynirci [9] propose an interactive approach that combines UTADIS and [11], assuming an underlying additive utility function. They find the priority of categories to classify all the alternatives. DM assigns alternatives to their categories, if it is feasible and they place all alternatives based on DM's past preferences. Buğdacı et al. [3] propose an interactive probabilistic sorting method. They calculate the probability for each unassigned alternative. They find the critical probability level. Unassigned alternative probability is compared with the critical probability level to assign alternatives to classes.

There are several studies that develop interactive algorithms for selection problem using the convex cones approach. Malakooti [13] and [14] propose heuristic and exact algorithms, respectively, to identify and eliminate inefficient alternatives, hence reducing the pairwise comparisons required. Karsu et al. [6] propose an interactive ranking method using convex cones.

In this study, we use an iterative algorithm, where, in each iteration, we calculate the likelihood that an alternative is preferred to another one for each pair. According to some function of these likelihood information, we select the pair to be demonstrated to the DM for comparison. According to DM's answer, we update the likelihood information in the next iteration.

3 Likelihood Computation Method

In this section we present our approach to compute the likelihood that DM will prefer an alternative to another. We first estimate the minimum and maximum value of each alternative given the gathered preference information through the iterations of the algorithm. We then present the computation of the likelihoods based on ranges defined by the minimum and maximum values.

3.1 Estimating Value Ranges

In order to calculate the likelihood that DM prefers an alternative to another one, we first compute the minimum and the maximum values each alternative may have. During the computations, we assume that DM has a linear value function. We solve the following model two times for each alternative: one for minimizing and one for maximizing the estimated value.

In this model, the parameter x_{iq} is the score of alternative i in criterion q. The decision variable w_q represents the weight of criterion q and μ_i is a decision variable representing the estimated value of alternative i for a linear value function.

Second constraint set presents the gathered preference information. After DM responses that she prefers alternative i to alternative k, a new constraint is added to this set.

$Maximum/Minimum\ Value\ Model$:

$$\begin{aligned}
Max/Min\ Z &= \mu_i \\
s.t. & \\
\mu_i &= \sum_{q=1}^{p} x_{iq} w_q, \forall i \\
\mu_i &\geq \mu_k, i \succ k \\
\sum_{q=1}^{p} w_q &= 1 \\
\mu_i &\geq 0, \forall i \\
w_q &\geq 0, \forall q
\end{aligned}$$

We denote the maximum and minimimum values of the above model for alternative i as $f_{max}(i)$ and $f_{min}(i)$, respectively.

3.2 Likelihood Estimation

In this section, we show how the estimated likelihood that each alternative is preferred to another one is calculated. We utilize a uniform probability distribution. Let $P(i,k)$ be the likelihood that alternative i is preferred to alternative k. Three cases are possible considering $f_{max}(i)$, $f_{min}(i)$, $f_{max}(k)$ and $f_{min}(k)$.

Case 1: If $f_{max}(i) \geq f_{max}(k) \geq f_{min}(i) \geq f_{min}(k)$

$$P(i,k) = \frac{f_{max}(i) - \left(\frac{f_{max}(k)+f_{min}(i)}{2}\right) + f_{min}(i) - f_{min}(k)}{f_{max}(i) - f_{min}(k)}$$

Case 2: If $f_{max}(i) \geq f_{max}(k)$ and $f_{min}(i) \leq f_{min}(k)$

$$P(i,k) = \frac{f_{max}(i) - \left(\frac{f_{max}(k)+f_{min}(k)}{2}\right)}{f_{max}(i) - f_{min}(i)}$$

Case 3: If $f_{min}(i) \geq f_{max}(k)$

$$P(i,k) = 1$$

3.3 Model for Finding the Alternatives Eliminated by Each Cone

Although we do not have DM's preference information, we can still compute the consequences of each possible answer of DM. For each possible answer, we will generate a convex cone and this cone will eliminate some alternatives. For this purpose we develop a mathematical programming model. We assume Cone(x_i, x_k) is generated when DM prefers alternative i to alternative k. The model checks if Cone(x_i, x_k) dominates alternative t for all possible (i,k,t) triplets.

We develop a mathematical programming model, that finds the alternatives eliminated by each cone (generated by a single pairwise comparison). In this model, there are $k, i, t = 1, \ldots m$ alternatives evaluated on $q = 1, \ldots, p$ criteria. The parameter x_{iq} is the score of alternative i on criterion q. We assume, for each criterion, higher score is better. There are two types of decision variables; ϵ_{ikt} and μ_{ikt}. The first type of decision variables are unrestricted in sign where the second type is nonnegative. We present the model below:

$Combined\ Cone\ Model$:

$$\begin{aligned}
Max\ Z &= \sum_{i=1}^{m} \sum_{\substack{k=1 \\ i \neq k}}^{m} \sum_{\substack{t=1 \\ t \neq i \\ t \neq k}}^{m} \epsilon_{ikt} \\
s.t. & \\
\mu_{ikt}(x_{kq} - x_{iq}) - \epsilon_{ikt} &\geq x_{tq} - x_{kq}, \forall q, i, k, t \\
\mu_{ikt} &\geq 0, \forall i, k, t
\end{aligned}$$

The objective function combines the objective functions of the individual models written for all possible (i,k,t) triplets as shown in Theorem 1. The constraint set includes the constraints of the individual models for all (i,k,t) triplets. If $\epsilon_{ikt} \geq 0$ in the solution of the model, x_t which is dominated by Cone(x_i, x_k), is eliminated. If $\epsilon_{ikt} < 0$ in the solution of the model, x_t which is non-dominated, is not eliminated by Cone(x_i, x_k). Note that this model can be decomposed into smaller models for each (i,k,t) triplets to check whether Cone(x_i, x_k) dominates alternative t or not.

Let $\epsilon^{\star}_{ikt}$ be the value of ϵ_{ikt} at the optimal solution. We compute $NE(i,k)$, the number of alternatives that will be eliminated by Cone(x_i, x_k) as follow:

$$NE(i,k) = \sum_{\substack{t=1, t\neq i, t\neq k\\ \epsilon^*_{ikt}\geq 0}}^{m} 1$$

4 Interactive Algorithm

In this section, we discuss the interactive algorithm. We present the steps of the algorithm and later discuss each step in detail.

Step 0. Eliminate the dominated alternatives.

Step 1. Solve combined cone model and compute initial $NE(i,k)$ values for each pair of alternatives (i,k).

Step 2. Find the minimum and the maximum scores of value functions for each alternative (in case of infeasibility, remove the oldest constraints one by one until feasibility is reached).

Step 3. Compute $P(i,k)$ values and $E[i,k]$, expected number of alternatives to be eliminated by the cone that will be generated depending on the response of the DM. This value is the weighted average of the number of alternatives that will be eliminated by Cone(x_i, x_k) and Cone(x_k, x_i) where the weights correspond to the likelihood.

Step 4. Pick the alternative pair (i,k) with the highest $E[i,k]$ value and ask DM to compare these alternatives. Assume that DM prefers alternative i to alternative k (otherwise swap i and k).

Step 5. Eliminate alternative k and other alternatives dominated by Cone(x_i, x_k), add a new constraint to model and update $NE(i,k)$ values.

Step 6. If there is only one alternative left, go to Step 7 otherwise go Step 2.

Step 7. Report the remaining alternative as the most preferred alternative and stop.

In Step 1, the algorithm first solves the combined cone model and computes $NE(i,k)$ values for each pair.

In Step 2, the minimum and maximum values of each alternative are determined by the mathematical model given in Section 3.1. We compute these values assuming that DM has a linear value function, however this assumption may not hold and the models may become infeasible. In such a case, as suggested in [12], we start removing constraints one by one starting from the oldest one until obtaining feasibility.

In Step 3, we compute the $P(i,k)$ values based on minimum and maximum scores obtained in Step 2. We also compute $E[i,k]$ values, the expected number of alternatives to be eliminated by asking the DM to compare alternatives i and k using the following equation:

$$E[i,k] = P(i,k) \times NE(i,k) + P(k,i) \times NE(k,i)$$

In this equation, we assume that the DM will prefer alternative i to alternative k with probability $P(i,k)$ and $NE(i,k)$ alternatives will be eliminated in this case. With probability $P(k,i)$, the DM will prefer alternative k to alternative i and $NE(k,i)$ alternatives will be eliminated. Hence, $E[i,k]$ gives us the expected number of alternatives that will be eliminated as a result of comparing alternatives i and k. In case the DM is indifferent between i and k, we select the pair with the next highest expected number of eliminated alternatives to be compared by the DM.

In Step 4, the algorithm picks the pair with the highest $E[i,k]$ value and asks the DM to prefer one alternative. Based on the gathered information and computed likelihoods, this pair is expected to eliminate the highest number of alternatives. Without loss of generality, let us assume DM prefers alternative i to alternative k.

In Step 5, the algorithm removes alternative k, generates Cone(x_i, x_k) and eliminates alternatives dominated by this cone. The following constraint representing the DMs response is added to the mathematical models given in Section 3.1:

$$\mu_i \geq \mu_k$$

The algorithm also updates $NE(i,k)$ values since some of the alternatives dominated by the corresponding cone may be already eliminated by Cone(x_i, x_k). Note that, this update does not require solving the combined model once again.

In Step 6, the algorithm checks the number of remaining alternatives and returns to Step 2 if there are more than one alternative. Otherwise, it reports the remaining alternative as the most preferred one in Step 7.

We assume there are m decision alternatives available. The algorithm removes at least one alternative at each iteration and terminates in at most $m-1$ iterations. At each iteration, the algorithm requires solving two LP problems for each remaining alternative and computing $NE(i,k)$ and $E[i,k]$ for each pair of remaining alternatives.

5 An Illustrative Example

In this section, we implement our algorithm on an example problem presented in [12]. There are nine alternatives evaluated on three criteria (u_1, u_2, u_3). It is assumed that the DM has an underlying quadratic value function as below:

$$-(u_1 - 66)^2 - (u_2 - 80)^2 - (u_3 - 75)^2$$

We use the value function only to simulate the DM's response to the pairwise comparisons. The scores of alternatives on each criteria and the values are given in Table 1.

Table 1. Scores of the alternatives

	Criteria			
Alternatives	u_1	u_2	u_3	Value
1	66	30	-12	-10069
2	48	60	12	-4693
3	36	12	72	-5533
4	24	66	66	-2041
5	60	20	-20	-12661
6	15	-15	75	-11626
7	30	30	15	-7396
8	20	80	40	-3341
9	0	0	0	-16381

In Step 0, we eliminate alternatives 5 and 9 since they are dominated by at least one of the alternatives. In Step 1, we obtain the number of eliminated alternatives by each cone. In Step 2, we find $f_{min}(i)$ and $f_{max}(i)$ values for all alternatives using the mathematical programming model given in Section 3.1. We report the initial values in Table 2. Note that the algorithm updates these values throughout the iterations.

In Step 3, the algorithm computes $P(i,k)$ and $E[i,k]$ values and in Step 4 it asks DM to compare the two alternatives with the high-

Table 2. Estimated value intervals of the alternatives (Initial)

	Estimated Values	
Alternatives	Minimum	Maximum
1	0	66
2	17.5	54
3	22	66.5
4	24	66
6	0	65.8
7	17.3	30
8	20	73.3

est $E[i,k]$ value. We report sample computation of these values at an intermediate iteration of the algorithm in Tables 3 and 4. In Table 3, we present three pairs of alternatives, each corresponding to a different case discussed in Section 3.2. Note that, with the introduction of DM preferences, the intervals of the estimated values are tighter compared to those values reported in Table 2. In Table 4, we present the computation of $E[i,k]$ values for the pairs $(3,1)$, $(1,2)$ and $(4,7)$.

Table 3. Sample computation of $P(i,k)$ values

	Utility Ranges			Utility Ranges			
i	Min.	Max.	k	Min.	Max.	Case	Likelihood
3	31.2	66.5	1	0	49.1	1	0.865
1	0	49.1	2	17.5	49.1	2	0.321
4	40.1	66	7	17.3	28.9	3	1.000

Table 4. Sample computation of $E[i,k]$ values

(i,k)	$NE(i,k)$	$NE(k,i)$	$P(i,k)$	$P(k,i)$	$E[i,k]$
(3,1)	2	1	0.865	0.135	1.865
(1,2)	1	3	0.321	0.679	2.357
(4,7)	3	1	1.000	0.000	3.000

After we get the preference information from DM, we generate the corresponding cone and eliminate alternatives dominated by this cone. The algorithm returns to Step 2 and continues until only one alternative is left, which is reported as the most preferred one.

6 Conclusion

In this study, we develop an interactive algorithm for the multiple criteria selection problem. The interactive algorithm asks DM to make pairwise comparisons and uses the responses of DM to eliminate alternatives by generating convex cones. Moreover, the algorithm utilizes the gathered information to detect the next pairwise comparison question via a likelihood computation.

We present an illustrative example for explaining the steps of the algorithm. As a further research, we plan to conduct computational experiments to compare the performance of the presented algorithm, its variations and other algorithms available in the literature.

A commonly used performance metric for such interactive algorithms is the average number of pairwise questions asked to DM in order to select the most preferred alternative. In general, the performance of a new algorithm is measured relative to other algorithms (available in the literature or variants of the new one) in terms of average questions asked. The comparison of our interactive algorithm with the algorithms existing in the literature is another further research direction.

REFERENCES

[1] Argyris N., Morton A., Figueria J.R. 2014. CUT: A Multicriteria Approach for Concavifiable Preferences, Operations Research, 62, 633-642

[2] Benyoucef L., Ding H., Xie X. 2003. Supplier Selection Problem: Selection Criteria and Methods, Institut National De Recherche En Informatique Et En Automatique, Feb 2003, 1-38.

[3] Buğdacı A., Köksalan M., Özpeynirci S., Serin Y. 2012. An Interactive Probabilistic Approach to Multi-Criteria Sorting, IIE Transactions, 45, 1048-1058.

[4] Ehrgott M. 2005. Multicriteria Optimization. Second edition. Springer, Berlin.

[5] Karsu Ö. 2013. Using Holistic Multicriteria Assessments: The Convex Cones Approach. Wiley Encyclopedia of Operations Research and Management Science. 114.

[6] Karsu Ö., Morton A., Argyris N. 2012. Incorporating Preference Information in Multicriteria Problems with Equity Concerns, Working Paper LSEOR 12.136, London School of Economics and Political Science.

[7] Köksalan M. 1989. Identifying and Ranking a Most Preferred Subset of Alternatives in the Presence of Multiple Criteria, Naval Research Logistics, 36, 359-372.

[8] Köksalan M., Karwan M.H., Zionts S. 1984. An Improved Method for Solving Multiple Criteria Problems Involving Discrete Alternatives, IEEE Transactions on Systems, Man, and Cybernetics, SMC-14, 24-34.

[9] Köksalan M., Özpeynirci S. 2009. An Interactive Sorting Method for Additive Utility Functions, Computers&Operations Research, 36, 2565-2572.

[10] Köksalan M., Taner O.V. 1989. An Approach for Finding the Most Preferred Alternative in the Presence of Multiple Criteria, European Journal of Operational Research, 60, 52-60.

[11] Köksalan M., Ulu C. 2003. An Interactive Approach for Placing Alternatives in Preference Classes, European Journal of Operational Research, 144, 429-439.

[12] Korhonen P., Wallenius J., Zionts S. 1984. Solving the Discrete Multiple Criteria Problem Using Convex Cones, Management Science, 30 (11), 1336-1345.

[13] Malakooti B. 1988. A Decision Support System and a Heuristic Interactive Approach for Solving Discrete Multiple Criteria Problems, IEEE Transactions on Systems, Man, and Cybernetics, 18(2), 273-284.

[14] Malakooti B. 1989. Theories and an Exact Interactive Paired-Comparison Approach for Discrete Multiple-Criteria Problems, IEEE Transactions on Systems, Man, and Cybernetics, 19(2), 365-378.

[15] Roy B. 1996. Multicriteria Methodology for Decision Aiding, Kluwer Academic Publisher.

[16] Steuer R. 1986. Multiple Criteria Optimization: Theory, Computation, and Application. John Wiley, New York, USA.

[17] Taner O.V., Köksalan M. 1991. Experiments and an Improved Method for Solving the Discrete Alternative Multiple-criteria Problem, Journal of the Operational Research Society, 42(5), 383-391.

FlowSort parameters elicitation: the case of interval sorting

Dimitri Van Assche[1], Yves De Smet[2]

Abstract. We consider the context of interval sorting i.e. the possible assignment of alternatives into several successive categories. We address the problem of finding the parameters of the FlowSort method using an existing categorization. This contribution constitutes an extension of an approach we have developed in the context of traditional sorting. It relies on the use of a dedicated Genetic Algorithm based on variations of search parameters. We show how to manage the problem of correct categorization prediction, which is more difficult, since ranges of categories are now considered. The method is tested on three different datasets for which an interval sorting has been generated with a particular instantiation of FlowSort.

1 Introduction

Multi-criteria decision aid (MCDA) has been an active research field for more than 40 years. In this context, possible decisions are simultaneously evaluated on multiple conflicting criteria. For instance, in the common example of buying a new car, one typically tries to minimize the cost and consumption while maximizing performances, comfort, etc. Obviously no real car would be the best on all those criteria. Therefore, the notion of optimal solution is most of time replaced by the idea of compromise solution [8].

In this paper, we will work with the well-known outranking method PROMETHEE [1]. We focus on the sorting problem, i.e. the assignment of alternatives into predefined categories. For instance, sorting countries into risk categories on the basis of economical, financial and political indicators. In this paper, we work with FlowSort, which is a natural extension of PROMETHEE for sorting problems [7].

In the context of FlowSort, the decision maker needs to give central, or limit, profiles defining each category and preference parameters characterizing each criterion. Here, we consider the problem inside out: based on an existing categorization, one tries to find the parameters of FlowSort, which allow to best replicate the existing categorization.

Let us point out that we have recently proposed a first contribution on the preference elicitation for FlowSort based on assignment examples [10]. This paper only considers traditional sorting problems: each alternative is assumed to belong to a unique category. In between, we have slightly improved the performances of the Genetic Algorithm we used. Furthermore, we propose an extension of this first work to deal with interval sorting. The idea is that an alternative may belong to different successive categories at the same time.

When describing the so-called ”sorting problematic”, researchers usually refer to the assignment of alternatives into pre-defined categories. These are defined as ”single” classes ranked from the worst to the best one. However, many methods, such as ELECTRE TRI or FLOWSORT, provide as outputs not only precise assignments but also interval assignments. In the context of ELECTRE TRI for instance, it is due to the fact that pessimistic and optimistic rules do not necessarily lead to the same outcome (same arguments can be provided for FlowSort). When the learning set has been obtained by the application of such methods, interval assignments are likely to appear. Therefore, restricting the inference approach (which aim is to replicate this input) to precise allocations seems to us a bit arbitrary. Moreover, in the context of preferences elicitation, one may easily imagine situations where a given decision maker provide statements like “this action does not belong to the first class”, “this action belongs to C_2 or C_3 but, due to imprecisions or the lack of additional information, I am not able to further refine this assertion”, etc. All these pieces of information help us to characterize, at least partially, the preferential model. As a consequence, the inference of parameters that replicate interval assignments seems to be quite natural in a multi-criteria context.

In section 2, we introduce PROMETHEE and FlowSort. In section 3, we describe the genetic algorithm we use to solve the related optimization problem. Then, in section 4, we illustrate the algorithm and its performances on different datasets for which a categorization has been computed with a particular instantiation of FlowSort.

2 PROMETHEE and FlowSort

In this section, we briefly present PROMETHEE[3] I and II as well as FlowSort. For additionnal information, we refer the interested reader to [3] for a detailed description of PROMETHEE and to [6] for FlowSort.

Let $A = \{a_1, a_2..., a_n\}$ be a set of n alternatives and let $F = \{f_1, f_2..., f_q\}$ be a family of q criteria. The evaluation of alternative a_i for criterion l will be denoted by a real value $f_l(a_i)$.

For each pair of alternatives, let's compute $d_l(a_i, a_j)$, the difference of a_i over a_j on criterion l.

$$d_l(a_i, a_j) = f_l(a_i) - f_l(a_j) \tag{1}$$

A preference function, denoted P_l, is associated to each criterion l. This function transforms the difference of alternatives' evaluations $d_l(a_i, a_j)$ into a preference degree of the first alternative over the second one for criterion l. Without loss of generality, we consider that criteria have to be maximized. P_l is defined as follows:

[1] Computer & Decision Engineering (CoDE), Université libre de Bruxelles, email: dvassche@ulb.ac.be

[2] Computer & Decision Engineering (CoDE), Université libre de Bruxelles, email: yvdesmet@ulb.ac.be

[3] Preference Ranking Organization METHod for Enrichment of Evaluations

$$P_l : \mathbb{R} \to [0,1] : x \to P_l(x) \quad (2)$$

such that:

- $\forall x \in \mathbb{R}^- : P_l(x) = 0$,
- $\forall x, y \in \mathbb{R}_0^+ : x \leq y \implies P_l(x) \leq P_l(y)$

There are different kinds of preference functions. Henceforth, we consider only the linear one (see figure 1) which is characterized by two parameters: an indifference and a preference threshold: q_l, p_l.

$$\pi_l(a_i, a_j) = P_l[d_l(a_i, a_j)] = \begin{cases} 0 & \text{if } d_l(a_i, a_j) \leq q_l \\ \frac{d_l(a_i, a_j) - q_l}{p_l - q_l} & \text{if } q_l < d_l(a_i, a_j) \leq p_l \\ 1 & \text{if } p_l < d_l(a_i, a_j) \end{cases} \quad (3)$$

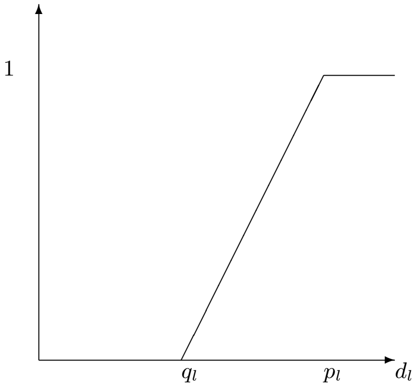


Figure 1. Linear preference function

Once $\pi_l(a_i, a_j)$ has been computed for all pairs of alternatives, we may define the aggregated preference degree of alternative a_i over a_j using the weights w_l associated to each criterion l. Weights are assumed to be positive and normalized.

$$\pi(a_i, a_j) = \sum_{l=1}^{q} w_l . \pi_l(a_i, a_j) \quad (4)$$

The last step consists in calculating the positive flow score denoted $\phi_A^+(a_i)$ and the negative flow score denoted $\phi_A^-(a_i)$ as follows:

$$\phi_A^+(a_i) = \frac{1}{n-1} \sum_{x \in A} \pi(a_i, x) \quad (5)$$

$$\phi_A^-(a_i) = \frac{1}{n-1} \sum_{x \in A} \pi(x, a_i) \quad (6)$$

We define the net flow score of a_i as the difference between the positive flow and negative flows of a_i:

$$\phi_A(a_i) = \phi_A^+(a_i) - \phi_A^-(a_i) \quad (7)$$

The PROMETHEE I ranking is obtained as the intersection of the rankings induced by ϕ^+ and ϕ^-. For an interpretation of the net flow scores, the interested reader is referred to [5]. Finally, a complete order, called PROMETHEE II, can be derived from the order induced by ϕ.

Based on PROMETHEE, FlowSort has been developed to address sorting problems [6]. Let $C = \{c_1, c_2 ..., c_k\}$ be a set of k ordered categories. We assume that $c_i \succ c_{i+1}$: c_i is preferred to c_{i+1}. Therefore C_1 is the best category and C_k is the worst one.

Categories are assumed to be represented by limit or central profiles. On the one hand, the idea of the limiting profiles is to define couples of values for each criterion, defining the lower and upper bounds of the considered category. Let us note that the profile defining the upper bound of category c_i is the same as the one defining the lower bound of category c_{i+1}. On the other hand, central profiles are defined using a single value for each criterion. This represents a kind of mean profile of the category. A common property is that the profiles of each category must dominate the profiles of the ones they are preferred to. In this work, we have chosen to work with central profiles.

Let's define $R = \{r_1, r_2 ..., r_k\}$, the set of central profiles representing the k categories. To identify the category of an alternative a_i, we define the subset $R_i = R \cup \{a_i\}$. Then, for each element x in the subset R_i, we compute its net flow score $\phi_{R_i}(x)$.

As in the nearest neighbor procedure, the category of alternative a_i is the one such that the profile has its net flow score the closest to the net flow score of a_i. More formally:

$$l^*(a_i) = \underset{l=1,2...,k}{argmin} |\phi_{R_i}(a_i) - \phi_{R_i}(r_l)| \quad (8)$$

Let us note that $(3 + k).q$ parameters have to be provided in order to instantiate FlowSort:

- $k.q$ values for the central profiles;
- $3.q$ values for the weights, indifference and preference thresholds.

In the case of the interval sorting problem, an alternative can be sorted in multiple consecutive categories. In this case, we use an extension of PROMETHEE I instead of PROMETHEE II. The upper and lower categories are determined using the positive and negative flow scores. As in the regular FlowSort method, the category of a_i is determined as the category of the profile having its positive, resp. negative, flow score the closest to the the one of the alternative a_i.[6]

$$l_+^*(a_i) = \underset{l=1,2...,k}{argmin} |\phi_{R_i}^+(a_i) - \phi_{R_i}^+(r_l)| \quad (9)$$

$$l_-^*(a_i) = \underset{l=1,2...,k}{argmin} |\phi_{R_i}^-(a_i) - \phi_{R_i}^-(r_l)| \quad (10)$$

If both values are equal, the categorization is precise. Otherwise, these values define the range of the categories.

3 Algorithms

The algorithms developed to learn the FlowSort parameters are the same as those presented in [10]. This approach is based on a dedicated genetic algorithm. Henceforth we only present the specific points that are dedicated to interval sorting i.e.:

- the definition of a distance measure in order to guide the optimization process;
- the evaluation of the correctness of a particular solution.

Compared to the previous approach [10], we have also completely changed the parameters optimization process. In [10], we have used iRace[4] in order to fine tune the parameters of the algorithm. For more information on the iRace procedure, we refer the interested reader to [2] [4]. We will describe the new procedure hereafter.

[4] Iterated Race for Automatic Algorithm Configuration

A distinctive feature of interval sorting is that we have to deal with two pieces of information: the upper and lower categories for each alternative. Let us note $c^+(a_i)$ the upper category and $c^-(a_i)$ the lower one. We have chosen to use the L_1 distance between the upper and lower category given as input c_r and the one given by the current parametrization of FlowSort c_f. Hence, s will denote the current parameters vector (in other words: a current solution). The distance is used to induce a higher penalty if there is a big difference between the prediction and the real category. Intuitively, the penalty associated to s is defined as follows:

$$f(s) = \sum_{a \in A} (|c_f^+(a) - c_r^+(a)| + |c_f^-(a) - c_r^-(a)|) \qquad (11)$$

For the sake of simplicity, we denote $c_f^+(a) = c_f^+(a, s)$. The same applies to c_r and c^-. The optimization problem is to find a parameters set that minimizes this distance.

The correctness defines how good a solution is with respect to the real categorization. The correctness of a single alternative is defined as the number of categories correctly predicted divided by the total range covered by the predicted and the real categories. The correctness of a solution s is defined as the sum of the correctness of all the alternatives:

$$\sum_{a \in A} \frac{max(min(c_f^+(a), c_r^+(a)) - max(c_f^-(a), c_r^-(a)), -1) + 1}{max(c_f^+(a), c_r^+(a)) - min(c_f^-(a), c_r^-(a)) + 1} \qquad (12)$$

The identification of the best possible solution regarding the correctness is based on a genetic algorithm. This algorithm has mainly two kinds of exploration: diversification with the mutation operator and intensification with the crossover operator. During the tests, we have observed that the algorithm should ideally enforce diversification after intensification and then go back to intensification, and so on. As a consequence, the idea we have applied is to force the parameters variation of the algorithm during the optimization process. There are 5 parameters: population size, mutation probability, gene mutation probability, crossover probability, gene crossover probability. The population size has been fixed to 1600 solutions. This value has been set after a set of trial and errors, and seems to work well in the considered examples. The 4 others parameters have values between 0 and 1. At each step of the optimization we change the values of those following a linear equation. When the value 0, or 1, is reached the coefficient is reversed. We paid attention to chose different coefficients, so that the period is different. This permits to have a lot of different combinations of intensification and combinations.

In table 1, we show the values we have chosen for p_{min} and p_{max} for each paramter.

parameter	p_{min}	p_{min}
mutation probability	0.1	0.9
gene mutation probability	0.25	0.99
crossover probability	0.1	0.9
gene crossover probability	0.25	0.99

Table 1. Values of p_{min} and p_{max} for each parameter.

With this new method, we have seen a slight improvement of the results compared to our previous work. We were able to increase the correctness of the prediction applied to the learning set. Unfortunately, this did not really improve the prediction rate on the test set using the parameters learned with the learning set.

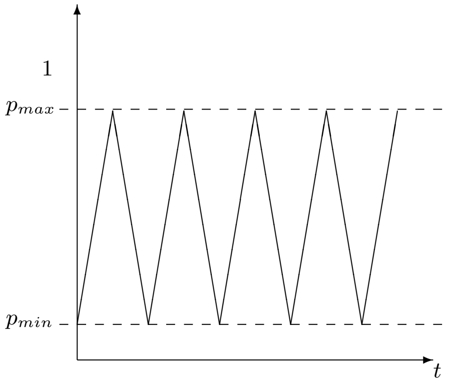


Figure 2. Varying parameters for the GA

4 Results

In [10], we have worked on 3 real datasets for the validation: CPU, BC and CEV. They come from the website of Marburg University[5]. These originally come from the UCI repository[6] and the WEKA machine learning toolbox[7]. Let us point out that these datasets have also been used by Sobrie [9] in the context of sorting but using a modified version of ELECTRE TRI.

To the best of our knowledge, there is no dataset for interval sorting. As a consequence, we decided to use the same 3 datasets but generating an interval categorization by using a random instantiation of FlowSort. Therefore we know an exact solution exists for the model's parameters. The properties of the datasets are in table 2.

dataset	#inst.	#crit.	#cat.	% imprecise cat.
CPU	209	6	4	40.67
BC	278	7	2	23.02
CEV	1728	6	4	16.43

Table 2. Datasets used for the tests

The testing procedure has been set as follows: each dataset has been divided in a learning set and a test set. Different sizes of learning set have been considered. Alternatives in the learning set have been randomly selected. Nevertheless, we forced the algorithm to select randomly at least one alternative from each category. For each learning and test set, the algorithm has been executed on the learning set to elicit the parameters. Then the values found have been evaluated on the test set. This operation has been executed 32 times for each learning set. For robustness' sake, the whole operation has been executed 10 times for each value of the learning set size. The maximum number of evaluations has been set to 2 500 000, and the population size to 1600. Results are available in table 3.

The correctness represents the accuracy of the prediction in the test set, and the learning set correctness represents the accuracy of

[5] http://www.uni-marburg.de/fb12/kebi/research/repository/monodata - September 2014

[6] http://archive.ics.uci.edu/ml/ - September 2014

[7] http://www.cs.waikato.ac.nz/ml/weka/datasets.html - September 2014

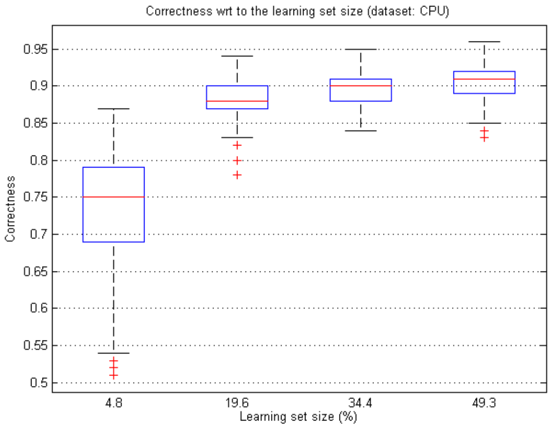


Figure 3. Correctness of the test set wrt LS size - CPU dataset

learning set's size	dataset	correctness	learning set correctness
5%	CPU	0.7337 ± 0.0705	1.0000 ± 0.0000
	BC	0.8827 ± 0.0337	0.9981 ± 0.0120
	CEV	0.8498 ± 0.0224	0.9227 ± 0.0383
20%	CPU	0.8798 ± 0.0245	0.9880 ± 0.0160
	BC	0.9463 ± 0.0209	0.9955 ± 0.0103
	CEV	0.8809 ± 0.0173	0.8554 ± 0.0338
35%	CPU	0.9004 ± 0.0215	0.9642 ± 0.0243
	BC	0.9579 ± 0.0210	0.9919 ± 0.0110
	CEV	0.8868 ± 0.0154	0.8395 ± 0.0277
50%	CPU	0.9065 ± 0.0228	0.9581 ± 0.0214
	BC	0.9747 ± 0.0163	0.9913 ± 0.0111
	CEV	0.8944 ± 0.0168	0.8309 ± 0.0252

Table 3. Results of the algorithm - correctness

the model on the learning set. From a global point of view, we can note that the correctness values are rather good. As expected, the correctness is increasing with the learning set's size. One can note that the learning set correctness is decreasing with the learning set size too. This is because it is much more complicated to have a perfect solution if the number of alternatives in the learning set increases. For the dataset CEV, the results show that the algorithm does not reach good level of learning set correctness (at least not as high as for the two other ones). The reason probably lies in the fact that this specific dataset is much bigger. We have noticed that the performances on the test set are better than on the learning set. Currently, we have no explanation for this effect. This will be further deepened in future works.

On figure 3, we show a boxplot of the evolution of correctness with respect to the learning set size (for the CPU dataset). We can note that a good level of correctness is already reached for a learning size of 20%.

Due to the introduction of the varying parameters for the GA, we did not need to fine tune the parameters anymore. The value of 1600 for the population size has been determined by trial and errors. Les us stress that the new method increases the running time. Nevertheless, it remains rather small. For instance, for the CPU dataset, the algorithm runs in about 5 minutes on a Intel i7-2640m with 8GB RAM, under Windows 8.1 with an implementation of the algorithm in Java 8. One advantage we have remarked during these first experiments is that the algorithm seems to be less stuck in local optima.

5 Conclusion

In this paper, we have addressed the question of preference elicitation in the context of interval sorting. To the best of our knowledge, this is the first attempt to solve such kind of problems. Furthermore, we have limited the analysis to a specific sorting method namely FlowSort. The approach is based on an extension of a method previously developed for sorting. It relies on the use of a dedicated genetic algorithm based on parameters variations. This has been illustrated on three real datasets. The validation was based on a learning set and test set created by a random instantiation of FlowSort. First experiments have shown that the algorithm runs quite fast and leads to good prediction values. A number of research questions are still to be addressed. Among others, we could investigate how an exact method could partly cover the elicitation process (typically the identification of weight values). From an algorithmic point of view, a detailed analysis of the heuristic is still to be done. More precisely, quantitative arguments have to be highlighted in order to confirm the added value of parameters variations. The use of benchmark datasets (that are not linked to a particular method like in this study) will certainly have an impact on the prediction quality. Nevertheless, the existence (or the creation) of such datasets is far from being obvious. Finally, the comparison between different sorting methods will probably lead to identify distinctive features that will be more appropriate to replicate particular categorizations.

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] Mauro Birattari, Zhi Yuan, Prasanna Balaprakash, and Thomas Stützle. F-race and iterated f-race: An overview. In *Experimental methods for the analysis of optimization algorithms*, pages 311–336. Springer, 2010.

[3] Jean-Pierre Brans and Bertrand Mareschal. Promethee methods. In *Multiple Criteria Decision Analysis: State of the Art Surveys*, volume 78 of *International Series in Operations Research & Management Science*, pages 163–186. Springer New York, 2005.

[4] Manuel López-Ibáñez, Jérémie Dubois-Lacoste, Thomas Stützle, and Mauro Birattari. The irace package, iterated race for automatic algorithm configuration. Technical Report TR/IRIDIA/2011-004, IRIDIA, Université libre de Bruxelles, Belgium, 2011.

[5] Bertrand Mareschal, Yves De Smet, and P Nemery. Rank reversal in the promethee ii method: some new results. In *Industrial Engineering and Engineering Management, 2008. IEEM 2008. IEEE International Conference on*, pages 959–963. IEEE, 2008.

[6] Philippe Nemery. *On the use of multicriteria ranking methods in sorting problems*. PhD thesis, PhD Thesis. Université libre de Bruxelles, 2008-2009, 2008.

[7] Philippe Nemery and Claude Lamboray. Flowsort: a flow-based sorting method with limiting or central profiles. *Top*, 16(1):90–113, 2008.

[8] Bernard Roy and Philippe Vincke. Multicriteria analysis: survey and new directions. *European Journal of Operational Research*, 8(3):207–218, 1981.

[9] Olivier Sobrie, Vincent Mousseau, and Marc Pirlot. Learning a majority rule model from large sets of assignment examples. In *Algorithmic Decision Theory*, pages 336–350. Springer, 2013.

[10] Dimitri Van Assche and Yves De Smet. Flowsort parameters elicitation based on classification examples. Technical Report TR/SMG/2014-003, SMG, CoDE, Université Libre de Bruxelles, Brussels, Belgium, June 2014.

On confident outrankings with multiple criteria of uncertain significance

Raymond Bisdorff
University of Luxembourg, FSTC/CSC/ILIAS
raymond.bisdorff@uni.lu

Abstract. When modelling preferences following the outranking approach, the sign of the majority margins do sharply distribute validation and invalidation of pairwise outranking situations. How can we be confident in the resulting outranking digraph, when we acknowledge the usual imprecise knowledge of criteria significance weights and a small majority margin? To answer this question, we propose to model the significance weights as random variables following more less widespread distributions around an average weight value that corresponds to the given deterministic weight. As the bipolarly valued random credibility of an outranking statement results from a simple sum of positive or negative independent and similarly distributed random variables, we may apply the CLT for computing likelihoods that a given majority margin is indeed positive, respectively negative.

Keywords: Multiple criteria decision aid; Uncertain criteria weights; Stochastic outranking relations; Confidence of the Condorcet outranking digraph.

Introduction

In a social choice problem concerning a very important issue like amending a country's Constitution, the absolute majority of voters is often not seen as sufficient for supporting a convincing social consensus. A higher majority of voters, two third or even three forth of them, may be required to support the bill in order to take effective decisions. Sometimes, even unanimity is required; a condition that, however, may generate in practice many indecisive situations. A similar idea is sometimes put forward in multiple criteria decision aiding in order to model global compromise preferences when the significance of the criterion are not known with sufficient precision. In his seminal work on the Electre I method (Roy [1], concerning a best unique choice problematique, Roy is clearly following this line of thought by proposing to choose a sufficiently qualified majority of criterial support before considering an outranking statement to be significant.

Following the SMAA approach (Tervonen et al. [2]), we are here proposing a different approach. The individual criteria significance weights are considered to be random variables. The bipolarly valued characteristic of the pairwise outranking situations (Bisdorff [3, 4]) appear hence to be sums of random variables of which we may assess the apparent likelihood of obtaining a positive weighted majority margin for each outranking situation. And depending on the seriousness of the decision issue, we may hence recommend to accept only those outranking statements that show a sufficiently high likelihood of 90% or 95%, for instance. We could also, in the limit accept only those statements which appear to be certainly supported by a weighted majority of criterial significance.

The paper is structured as follows. A first section is concerned with how to model the uncertainty we face for assessing precise numerical criteria significance weights. The second section illustrates how the likelihood of outranking situations may be estimated. The third section introduces the concept of confidence level of the valued outranking digraph, followed by short last section devoted to an illustrative example of confident best choice recommendation.

1 Modelling uncertain criteria significances

We have already extensively discussed some time ago (see Bisdorff [5]) the operational difficulty to numerically assess with sufficient precision the actual significance that underlies each criterion in a multiple criteria decision aid problem. Even, when considering that all criteria are equi-significant, it is not clear how precisely (how many decimals ?) such a numerical equality should be taken into account when computing the outranking characteristic values. In case of unequal significance of the criteria, it is possible to explore the stability of the Condorcet digraph with respect to the ordinal criteria significance structure (Bisdorff [6, 7]). One may also use indirect preferential observations for assessing via linear programming computations apparent significance ranges for each criterion (Dias [8]).

Here, we propose instead to consider the significance weights of a family F of n criteria to be independent random variables W_i, distributing the potential significance weights of each criterion $i = 1, ..., n$ around a mean value $E(W_i)$ with variance $V(W_i)$.

Choosing a specific stochastic model of uncertainty may be application specific. In the limited scope of this paper, we will illustrate the consequence of this design decision on the resulting outranking modelling with four slightly different models for taking into account the uncertainty with which we know the numerical significance weights: *uniform*, *triangular*, and two models of *Beta* laws, one more widespread and, the other, more concentrated. When considering that the potential range of a signficance weight is distributed between 0 and two times its mean value, we obtain the following random variates:

1. A continuous *uniform* distribution on the range 0 to $2 * E(W_i)$. Thus $W_i \sim \mathcal{U}(0, 2E(W_i))$ and $V(W_i) = \frac{1}{3}E(W_i)^2$;
2. A symmetric *beta*(a, b) distribution with, for instance, parameters $a = 2$ and $b = 2$. Thus, $W_i \sim \mathcal{B}eta(2,2) \times 2E(W_i)$ and $V(W_i) = \frac{1}{5}E(W_i)^2$.
3. A symmetric *triangular* distribution on the same range with mode $E(W_i)$. Thus $W_i \sim \mathcal{T}r(0, 2E(W_i), E(W_i))$ with $V(W_i) = \frac{1}{6}E(W_i)^2$;
4. A narrower *beta*(a, b) distribution with for instance parameters $a = 4$ and $b = 4$. Thus $W_i \sim \mathcal{B}eta(4,4) \times 2E(W_i)$, $V(W_i) = \frac{1}{9}E(W_i)^2$

It is worthwhile noticing that these four uncertainty models all admit the same expected value, $E(W_i)$, however, with a respective variance which goes decreasing from 1/3, to 1/9 of the square of $E(W_i)$ (see Fig. 1).

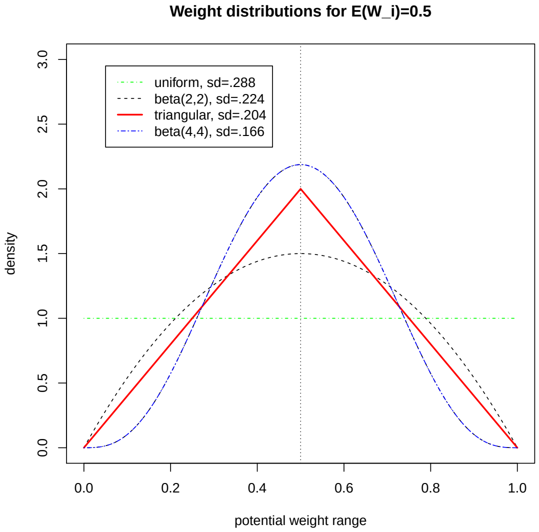


Figure 1. Four models of uncertain significance weights

We will limit in the sequel our attention to the triangular random model and explore now, without loss of generality, the resulting uncertainty we are going to model into the valued outranking digraph.

2 Likelihood of "*at least as good as*" situations

Let $A = \{x, y, z, ...\}$ be a finite set of n potential decision actions, evaluated on $F = \{1, ..., m\}$, a finite and coherent family of m performance criteria. On each criterion i in F, the decision actions are evaluated on a real performance scale $[0; M_i]$, supporting an upper-closed indifference threshold ind_i and a lower-closed preference threshold pr_i such that $0 \leqslant ind_i < pr_i \leqslant M_i$. The marginal performance of object x on criterion i is denoted x_i. Each criterion i is thus characterizing a marginal *double threshold* order $\succcurlyeq_i$ on A (see Fig. 2):

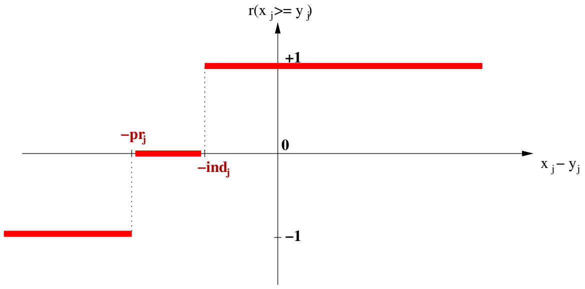


Figure 2. Characteristic function of marginal "*at least as good as*" statement

$$r(x \succcurlyeq_i y) = \begin{cases} +1 & \text{if} \quad x_i - y_i \geqslant -ind_i \\ -1 & \text{if} \quad x_i - y_i \leqslant -pr_i \\ 0 & \text{otherwise.} \end{cases} \qquad (1)$$

+1 signifies x is *performing at least as good as* y on criterion i,
−1 signifies that x is *not performing at least as good as* y on criterion i.
0 signifies that it is *unclear* whether, on criterion i, x is performing at least as good as y.

Each criterion $i \in F$ contributes the random significance W_i of his "*at least as good as*" characterization $r(\succcurlyeq_i)$ to the global characterization $\tilde{r}(\succcurlyeq)$ in the following way:

$$\tilde{r}(x \succcurlyeq y) \;=\; \sum_{i \in F} \left[\, W_i \cdot r(x \succcurlyeq_i y)\, \right] \qquad (2)$$

Thus, $\tilde{r}(x \succcurlyeq y)$ becomes a simple sum of positive or negative independent random variables with known means and variances where $\tilde{r} > 0$ signifies x is *globally performing at least as good as* y, $\tilde{r} < 0$ signifies that x is *not globally performing at least as good as* y, and $\tilde{r} = 0$ signifies that it is *unclear* whether x is globally performing at least as good as y.

From the Central Limit Theorem (CLT), we know that such a sum (Eq. 2) leads, with m getting large, to a Gaussian distribution Y with $E(Y) = \sum_i E(W_i) \times r(x \succcurlyeq y)$ and $V(Y) = \sum_i V(W_i) \times |r(x \succcurlyeq y)|$. And the likelihood of *validation*, respectively *invalidation* of an "*at least as good as*" situation, denoted $lh(x \succcurlyeq y)$, may be assessed as follows:

$$lh(x \succcurlyeq y) \;=\; \begin{cases} 1.0 - P(Y \leqslant 0.0) & \text{if } E[\tilde{r}(x \succcurlyeq y)] > 0, \\ P(Y \leqslant 0.0) & \text{otherwise.} \end{cases} \qquad (3)$$

Example 2.1. Let us consider two decision alternatives x and y being evaluated on a family of 7 equi-significant criteria, such that four out of the seven criteria positively support that x outranks y, and three criteria support that x does not outrank y. In this case, $\tilde{r}(x \succcurlyeq y) = 4w - 3w = w$ where $W_i = w$ for $i = 1, ..., 7$ and the outranking situation is positively validated. Suppose now that the significance weights W_i appear only more or less equivalent and let us model this numerical uncertainty with independent triangular laws: $W_i \sim \mathcal{T}r(0, 2w, w)$ for $i = 1, ...7$. The expected credibility of the outranking situation, $E(\tilde{r}(x \succcurlyeq y)) = 4w - 3w = w$, will remain the same, however with a variance of $7 \times \frac{1}{6}w^2$. If we take a unit weight $w = 1$, we hence obtain a standard deviation of 1.08. Applying the CLT we notice that, under the given hypotheses, the likelihood $lh(x \succcurlyeq y)$ of obtaining a positive majority margin will be about $1.00 - P(\frac{\tilde{r}-1}{1.08} \leqslant 0.0) \approx 83\%$. A Monte Carlo simulation with 10 000 runs empirically confirms the effective convergence to a Gaussian: $\tilde{r}(x \succcurlyeq y) \rightsquigarrow \mathcal{N}(1.03, 1, 089)$ (see Figure 3), with an empirical probability of observing a negative majority margin $P(\tilde{r}(x \succcurlyeq y) \leqslant 0.0)$ of indeed about 17%.

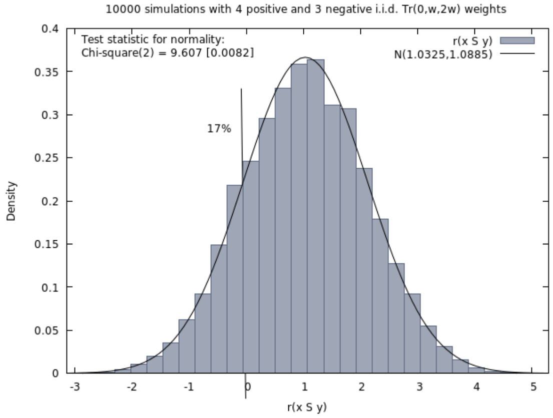


Figure 3. Distribution of outranking credibility $\tilde{r}(x \succcurlyeq y)$

Example 2.2. The second example concerns two decision alternatives a_1 and a_2 that are evaluated on a family of 7 criteria, denoted g_i of unequal significance weights w_i for $i = 1, ..., 7$ (see Tab. 1). The performances on the seven criteria are measured on a rational scale from 0 (worst) to 100 points (best). Let us suppose that both decision alternatives are evaluated as shown in Tab. 1. A performance difference of 10 points or less is considered insignificant, whereas a difference of 20 points and more is considered to be significant.

Table 1. Pairwise comparison of two decision alternatives

g_i w_i	g_1 7	g_2 8	g_3 3	g_4 10	g_5 1	g_6 9	g_7 7
a_1	14.1	71.4	87.9	38.7	26.5	93.0	37.2
a_2	64.0	87.5	67.0	82.2	80.8	80.8	10.6
$a_1 - a_2$	-49.9	-16.1	+20.9	-43.5	-54.3	+12.2	26.5
$r(\succcurlyeq_i)$	-1	0	$+1$	-1	-1	$+1$	$+1$

The overall deterministic outranking credibility $r(a_1 \succcurlyeq a_2)$ (see [4]) is given as follows:

$$r(a_1 \succcurlyeq a_2) = \sum_{i=1}^{7} r(a_1 \succcurlyeq_i a_1) \times w_i \quad (4)$$

$$= -7 + 0 + 3 - 10 - 1 + 9 + 7 = +1 \quad (5)$$

The outranking situation "$(a_1 \succcurlyeq a_1)$" is thus positively validated (see Eq. 5). However, in case the given criteria significance weights (see Tab. 1) are not known with certainty, how confident can we be about the actual positiveness of $\tilde{r}(a_1 \succcurlyeq a_2)$? If we suppose now that the random significance weights W_i are in fact independently following a triangular continuous law on the respective ranges 0 to $2w_i$, the CLT approximation will make $\tilde{r}(a_1 \succcurlyeq a_2)$ tend to a Gaussian distribution with mean equal to $E(\tilde{r}(x \succcurlyeq y)) = +1$ and standard deviation equal to $\sqrt{\sum_i 1/6E(W_i)^2} = 6.94$. The likelihood of $r(a_1 \succcurlyeq a_2) > 0.0$ equals thus approximately $1.0 - P(\frac{z-1}{6.94} \leqslant 0.0) = 1.0 - 0.443 \approx 55.7\%$, a result we can again empirically verify with a Monte Carlo sampling of 10 000 runs (see Fig. 4). Under the given modelling of the uncertainty in the setting of the criteria significance weights, the credibility of the outranking situation between alternatives a_1 and a_2 is neither convincingly positive, nor negative. The given relational situation may, hence, neither confidently be validated, nor, confidently invalidated.

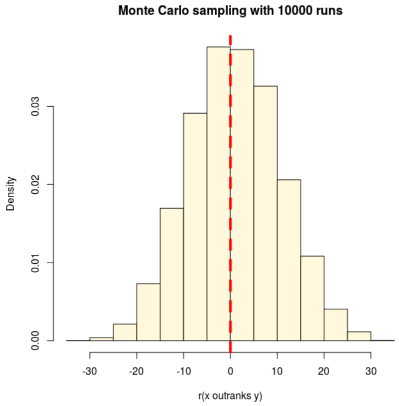


Figure 4. Distribution of outranking credibility $\tilde{r}(a_1 \succcurlyeq a_2)$

3 Confidence level of outranking situations

Following the classic outranking difinition (see Roy [1], Bisdorff [4]), we may say from an epistemic point of view, that decision action x *outranks* decision action y, denoted $x \succsim y$), if

1. a *confident majority* of criteria **validates** a global outranking situation between x and y, and
2. *no considerably less performing* is observed on a discordant criterion.

Dually, decision action x *does not outrank* decision action y, denoted $(x \not\succsim y)$, if

1. *a confident majority* of criteria **invalidates** a global outranking situation between x and y, and
2. *no considerably better performing* situation is observed on a concordant criterion.

On a criterion i, we characterize a *considerably less performing* situation, called *veto* and denoted $\lll_i$, as follows:

$$r(x \lll_i y) = \begin{cases} +1 & \text{if} \quad x_i + v_i \leqslant y_i \\ -1 & \text{if} \quad x_i - v_i \geqslant y_i \\ 0 & \text{otherwise.} \end{cases} \quad . \qquad (6)$$

where v_i, with $M_i \geqslant v_i > pr_i$, represents a lower-closed veto discrimination threshold. A corresponding dual *considerably better performing* situation, called *counter-veto* and denoted $\ggg_i$, is similarly characterized as:

$$r(x \ggg_i y) = \begin{cases} +1 & \text{if} \quad x_i - v_i \geqslant y_i \\ -1 & \text{if} \quad x_i + v_i \leqslant y_i \\ 0 & \text{otherwise.} \end{cases} \quad . \tag{7}$$

A global *veto*, or *counter-veto* situation is now defined as follows:

$$r(x \lll y) \quad = \quad \varovee_{i \in F} r(x \lll_i y) \tag{8}$$

$$r(x \ggg y) \quad = \quad \varovee_{j \in F} r(x \ggg_j y) \tag{9}$$

where $\varovee$ represents the epistemic polarising ([9]) or symmetric maximum ([10]) operator:

$$r \varovee r' = \begin{cases} \max(r, r') & \text{if} \quad r \geqslant 0 \wedge r' \geqslant 0, \\ \min(r, r') & \text{if} \quad r \leqslant 0 \wedge r' \leqslant 0, \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

We observe the following semantics:

1. $r(x \lll y) = 1$ iff there exists a criterion i such that $r(x \lll_i y) = 1$ and there does not exist otherwise any criterion $j \in F$ such that $r(x \ggg_j y) = 1$.
2. Conversely, $r(x \ggg y) = 1$ iff there exists a criterion i such that $r(x \ggg_i y) = 1$ and there does not exist otherwise any criterion j such that $r(x \lll_j y) = 1$.
3. $r(x \ggg y) = 0$ if either we observe no very large performance differences or we observe at the same time, both a very large positive and a very large negative performance difference.

It is worthwhile noticing that $r(\not\lll)^{-1}$ is identical to $r(\ggg)$, both $\lll$ and $\ggg$ being, by construction, codual relations one to another.

The deterministic outranking characteristic $r(\succsim)$ may hence be defined as follows:

$$r(x \succsim y) = r(x \succcurlyeq y) \varovee_{i \in F} \big[- r(x \lll_i y) \big] \tag{11}$$

And in particular,

1. $r(x \succsim y) = r(x \succcurlyeq y)$ if no very large positive or negative performance differences are observed,
2. $r(x \succsim y) = 0$ if a veto and a counter-veto situation are conjointly occurring;
3. $r(x \succsim y) = 1$ if $r(x \succcurlyeq y) \geqslant 0$ and $r(x \ggg y) = 1$,
4. $r(x \succsim y) = -1$ if $r(x \succcurlyeq y) \leqslant 0$ and $r(x \lll y) = 1$.

When considering now the criteria significance weights to be random variates, $r(x \succsim y)$ becomes a random variable via the random characteristic $\tilde{r}(x \succcurlyeq y)$.

$$\tilde{r}(x \succsim y) = \tilde{r}(x \succcurlyeq y) \varovee_{i \in F} \big[- r(x \lll_i y) \big] \tag{12}$$

In case 1. we are back to the unpolarised "*at least as good as*" situation discussed in the previous section. In case 2., the resulting constant indeterminate outranking characteristic value 0 is in fact independent of any criterion significance. Only cases 3. and 4. are of interest here. If $E(\tilde{r}(x \succcurlyeq y)) \geqslant 0$, we are in case 3. where strictly negative characteristics will be given the indeterminate characteristic 0, and the others, a polarised $+1$ value. Similarly, if $E(\tilde{r}(x \succcurlyeq y)) \leqslant 0$ we are in case 4., strictly positive characteristics $r(x \succcurlyeq y) > 0$ will be given the indeterminate value 0, and the others, the polarised -1 value.

By requiring now a certain level α of likelihood for effectively validating all pairwise outranking situations, we may thus enforce the actual confidence we may have in the valued outranking digraph. For any outranking situation $(x \succsim y)$ we obtain:

$$\hat{r}_\alpha(x \succsim y) \;=\; \begin{cases} E\big[\tilde{r}(x \succsim y)\big] & \text{if } lh(x \succcurlyeq y) \geqslant \alpha, \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

If, for instance, we would require that an outranking situation $(x \succsim y)$, to be validated, respectively invalidated, must admit a likelihood $lh(x \succcurlyeq y)$ of $\alpha = 90\%$ or more, any positively or negatively polarising of the "at least as good as" statement will only occur in case of sufficient likelihood. Noticing that $E\big[\tilde{r}(x \succsim y)\big] = r(x \succsim y)$, we safely preserve, hence, in our stochastic modelling, all the nice structural properties of the deterministic outranking relation (see Eq. 11), like *weak completeness* and *coduality*, that is the dual of the outranking relation ($\not\succsim$) corresponds to the asymmetric part ($\precnsim$) of its converse relation (see Bisdorff [4]).

Example 3.1. We may illustrate our uncertainty modelling approach with a small random performance tableau (see Tab. 2) showing the evaluations of seven decision alternatives on the same family of performance criteria we used for Example 2.2. To operate with a full fledged outranking model, we furthermore consider that a performance difference of 80 points and more will trigger a veto or counter-veto situation (see [4]).

Table 2. Random performance tableau

g_i	w_i	a_1	a_2	a_3	a_4	a_5	a_6	a_7
g_1	7	14.1	64.0	73.4	36.4	30.6	85.9	97.8
g_2	8	71.4	87.5	61.9	84.7	60.4	54.5	45.8
g_3	3	87.9	67.0	25.2	34.2	87.3	43.1	30.4
g_4	10	38.7	82.2	94.1	86.1	34.1	97.2	72.2
g_5	1	26.5	80.8	71.9	21.3	56.4	88.1	15.0
g_6	9	93.0	80.8	23.2	57.2	81.4	16.6	93.0
g_7	7	37.2	10.6	64.8	98.9	69.9	24.7	13.6

Thresholds: $ind_i = 10.0$, $pr_i = 20$, and $v_i = 80$ for $i \in F$.

When using the deterministic criteria significance weights shown in Tab. 2, we obtain the bipolarly valued outranking relation shown in Tab. 3. We recover there the weakly positive credibility ($r(a_1 \succsim a_2) = +1/45$) of the outranking situation between alternative a_1 and alternative a_2 discussed in Example 2.2. Notice also the slightly negative credibility ($-5/45$) of the outranking situation between alternative a_1 and a_3. Notice, furthermore the veto and counter-veto situations we observe when comparing alternatives a_1 and a_7, a_2 and a_4, as well as, a_4 and a_7. How confident are all these pairwise preferential situations when the significance weights are not precisely given? Assuming that the criteria significance weights w_i are in fact random variates distributed following independent triangular laws $\mathcal{T}(0, 2w_i, w_i)$ for $i = 1, \ldots, 7$, we obtain

Table 3. Deterministic credibility of $(x \succsim y)$

$r(\succsim) \times 45$	a_1	a_2	a_3	a_4	a_5	a_6	a_7
a_1	-	**+1**	**−5**	−11	+22	+9	**0**
a_2	+16	-	+21	**0**	+25	+14	+22
a_3	+21	+5	-	−3	+21	+34	+13
a_4	+21	**+45**	+29	-	+19	+19	**+45**
a_5	+28	−7	+10	−5	-	+9	+2
a_6	+6	+5	+31	−3	+7	-	+20
a_7	**+45**	+11	+1	**0**	+15	+13	-

Table 4. CLT likelihood of the $(x \succcurlyeq y)$ situations

lh	a_1	a_2	a_3	a_4	a_5	a_6	a_7
a_1	-	.56	.74	.94	1.0	.88	.92
a_2	.99	-	1.0	**1.0**	1.0	.99	1.0
a_3	1.0	.74	-	.65	1.0	1.0	.95
a_4	1.0	**.74**	1.0	-	.99	1.0	.95
a_5	1.0	.82	.90	.74	-	.88	.62
a_6	.83	.74	1.0	.65	.82	-	1.0
a_7	.85	.95	.56	.78	.98	.97	-

the CLT likelihoods shown Tab. 4. If we, now, require for each "*at least as good as*" situation $(x \succcurlyeq y)$ to admit a likelihood of 90% and more for convincingly validating, respectively invalidating, the corresponding outranking statement $(x \succsim y)$, we obtain the result shown in Tab. 5.

Table 5. 90% confident outranking characteristics (×45)

$\hat{r}_{90\%}(x \succsim y)$	a_1	a_2	a_3	a_4	a_5	a_6	a_7
a_1	-	0	0	−11	+22	0	0
a_2	+16	-	+21	0	+25	+14	+22
a_3	+21	0	-	0	+21	+34	+13
a_4	+21	0	+29	-	+19	+19	+45
a_5	+28	0	+10	0	-	0	0
a_6	0	0	+31	0	0	-	+20
a_7	0	+11	0	0	+15	+13	-

We notice there that, for instance, the outranking situations $(a_1 \succsim a_2)$ and $(a_1 \not\succsim a_3)$, with likelihoods 56%, resp. 73% – lower than 90%– are both put to doubt. Similarly, the +45 polarised outranking situation $(a_7 \succsim a_1)$ appears not confident enough. Same happens to +45 polarised situation $(a_4 \succsim a_2)$. Whereas situation $(a_4 \succsim a_7)$ remains confidently polarised to +1. In total 16 pairwise outranking statements, out of the potential 7 × 6 statements, are thus considered not confident enough. At required confidence level of 90%, their credibility $\hat{r}_{90\%}(x \succsim y)$ is put to the indeterminate value 0. It is worthwhile noticing that all outranking situations, showing a majority margin between ±9/45 (between 40 and 60%) are thus not confident enough and consequently put to doubt (characteristic value 0).

4 Exploiting the confident outranking digraph

Many MCDA decision aiding problematiques like best choice, ranking, sorting, or clustering recommendations based on pairwise outranking situations, rely on majority cuts of the corresponding valued outranking digraph (see [11, 12, 13]).

Example 4.1. The previous example 3.1 gives the hint how we may appreciate the very confidence we may have in a given majority when the criteria significance weights are not precisely given. We may, for instance, notice that alternative a_4 gives apparently the only Condorcet winner in the deterministic outranking digraph and will hence be recommended in the Rubis decision aid approach as best choice (see [13]). In the 90% confident outranking digraph, however, alternatives a_2 and a_4 both give two equivalent weak Condorcet winners, and may, hence, be both recommended as potential Rubis best choice candidates; a recommendation more convincingly supported than the deterministic one, when considering in fact the excellent performances of alternative a_2 compared to a_4 (see Tab. 6).

Table 6. Pairwise comparison of alternatives a_4 and a_2

g_i w_i	g_1 7	g_2 8	g_3 3	g_4 10	g_5 1	g_6 9	g_7 7
a_4	36.5	84.7	34.2	86.1	21.3	57.2	98.9
a_2	60.0	87.5	67.0	82.2	80.8	80.8	10.6
$-$	-27.5	-2.8	-32.8	+3.8	-59.2	-23.6	+88.8
$\succcurlyeq_j$	−1	+1	−1	+1	−1	−1	+1
$\ggg_j$	0	0	0	0	0	0	+1

$$\big[r(a_3 \succcurlyeq a_2) = +5 \;\wedge\; r(a_4 \ggg a_2) = +1\big] \Rightarrow r(a_4 \succsim a_2 = +45;$$
$$\big[lh_{.90}(a_4 \succcurlyeq a_2) \;=\; .74\big] \;\Rightarrow\; \hat{r}(a_4 \succsim a_2) \;=\; 0.$$

Being confidently *at least a s good as* alternative a_4 ($lh(a_2 \succcurlyeq a_4) = 100\%$, see Tab. 4), alternative a_2 shows four excellent performances over 80.0, whereas alternative a_4 only shows three such high evaluations. The actual difference between the *deterministic* and the *confident* best choice recommendation stems in fact from the not confident enough polarisation of the counter-veto affecting the performance comparison between a_4 and a_2 ($lh(a_4 \succcurlyeq a_2) = 74\% \;<\; \alpha = 90\%$, see Tab. 4). Hence, alternative a_4 does no more appear alone as the Condorcet winner. Both, alternatives a_2 and a_4 appear as *confident weak Condorcet* winners, hence their joint recommendation as *confident best choice* candidates.

Knowing a priori the distribution of the significance weight of each criterion will genuinely be sufficient in practice for computing, with the so given means and variances, the CLT based likelihood of the fact that a bipolar outranking characteristics $r(x \succsim y)$ is positively validating, respectively negatively invalidating, the outranking situation "$(s \succsim y)$". The quality of the CLT convergence will, however, depend, first, on the number of effective criteria, i.e. non abstaining ones, involved in each pairwise comparison and, secondly, on the more or less differences in shape of the individual significance weight distributions. Therefore, with tiny performance tableaux, less than 25 decision actions and less than 10 criteria, we may estimate more precisely the actual likelihood of all pairwise outranking situations with a Monte Carlo (MC) simulation consisting of a given number of independent runs. Indeed, the present computational power available, even on modest personal computers, allow us to sufficiently sample a given outranking digraph construction.

Example 4.2. If we sample, for instance, 10 000 MC simulations of the previous outranking relation (see Tab. 3), by keeping the same uncertainty modelling of the criteria significances with random weights distributed like $\mathcal{T}(0, 2w_i, w_i)$, we obtain same empirical likelihoods (see Tab. 7).

Table 7. Empirical likelihoods of $(x \succcurlyeq y)$ with a MC sampling of 10 000 runs

p-value	a01	a02	a03	a04	a05	a06	a07
a01	-	.55	.74	.95	1.0	.88	.92
a02	.99	-	1.0	1.0	1.0	.99	1.0
a03	1.0	.74	-	.65	1.0	1.0	.96
a04	1.0	.75	1.0	-	1.0	1.0	.96
a05	1.0	.82	.90	.75	-	.88	.61
a06	.83	.74	1.0	.65	.82	-	1.0
a07	.85	.96	.55	1.0	.99	.97	-

We may thus verify again the very accurate convergence (in the order of $\pm 1\%$) of the CLT likelihoods, a convergence we already observed in Example 2.2, even with a small number of criteria.

Conclusion

In this paper we illustrate some simple models for tackling uncertain significance weights: uniform, triangular and beta laws. Applying the Central Limit Theorem, we are able to compute under these uncertainty models the actual likelihood of any pairwise *at least as good as* situations. This operational result, by adequately handling potential veto and counter-veto situations, allows to enforce a given confidence level on the corresponding outranking situations. On a small illustrative best choice problem, we eventually show the pragmatic decision aid benefit one may expect from exploiting a confident versus a classic deterministic outranking digraph. Acknowledging this operational benefit, one may finally be tempted to extend the uncertainty modelling, as in the SMAA approach, to the marginal performances. This is however, not needed, as traditionally the performance discrimination thresholds proposed in the outranking approach may well take care of any imprecision and uncertainty at this level.

References

[1] Roy B (1985) Méthodologie Multicritère d'Aide à la Décision. Economica Paris 1, 3

[2] Tervonen, T. and Figueira, J. R. (2008). A survey on stochastic multicriteria acceptability analysis methods. Journal of Multi-Criteria Decision Analysis (Wiley) 15(1-2): 1-14 1

[3] Bisdorff, R. (2000). Logical foundation of fuzzy preferential systems with application to the electre decision aid methods. Computers and Operations Research (Elsevier) 27:673-687 1

[4] Bisdorff, R. (2013) On polarizing outranking relations with large performance differences. Journal of Multi-Criteria Decision Analysis (Wiley) 20:3-12 1, 3, 4

[5] 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

[6] Bisdorff, R., P. Meyer and Th. Veneziano (2009). Inverse analysis from a Condorcet robustness denotation of valued outranking relations. In F. Rossi and A. Tsoukiás (Eds.), Algorithmic Decision Theory. Springer-Verlag Berlin Heidelberg, LNAI 5783: 180-191 1

[7] Bisdorff, R., P. Meyer and Th. Veneziano (2014). Elicitation of criteria weights maximising the stability of pairwise outranking statements. Journal of Multi-Criteria Decision Analysis (Wiley) 21: 113-124 1

[8] Dias, L. C. (2002). Exploring the Consequences of Imprecise Information in Choice Problems Using ELECTRE. In Bouyssou, D. et al. (eds) Aiding Decisions with Multiple Criteria (Springer ORMS) 44: 175-193 1

[9] Bisdorff, R. (1997). On computing fuzzy kernels from l-valued simple graphs. In 5th European Congress on Intelligent Techniques and Soft Computing EUFIT'97, 1: 97- 103 4

[10] Grabisch M., J.-L Marichal, R. Mesiar, and E. Pap (2009). Aggregation functions. Encyclopedia of Mathematics and its Application. Cambridge University Press 4

[11] Bisdorff, R. (2002), Electre like clustering from a pairwise fuzzy proximity index. European Journal of Operational Research EJOR (Elsevier) 138/2: 320-331 5

[12] Bisdorff, R., M. Pirlot and M. Roubens (2006). Choices and kernels from bipolar valued digraphs. European Journal of Operational Research (Elsevier) 175: 155-170 5

[13] Bisdorff, R., P. Meyer and M. Roubens (2008). RUBIS: a bipolar-valued outranking method for the choice problem. 4OR - A Quarterly Journal of Operations Research (Springer-Verlag) 6 (2): 143-165 5