

Approximating incompletely defined utility functions of qualitative multi-criteria modeling method DEX

Matej Mihelčić^{1,2} · Marko Bohanec³

Published online: 2 August 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract Decision analysis is aimed at supporting people who make decisions in order to satisfy their needs and objectives. The method called DEX is a qualitative multi-criteria decision analysis approach that provides support to decision makers in evaluating and choosing decision alternatives by using discrete attributes and rule-based utility functions. In this work, we extend our previous efforts of approximating complete, monotone DEX utility functions with methods Direct marginals, UTADIS and Conjoint analysis to incompletely defined utility functions. The experiments are performed both on functions obtained by solving real world decision making problems and on artificially created ones. The results show that all three methods provide accurate approximations of incompletely defined DEX utility functions, when the evaluation is done only on rules present in these incompletely defined functions. Among the three methods, the Conjoint analysis method generally has the best performance, however it is closely followed by the Direct marginals method. The Conjoint analysis method also achieves a better performance in approximating fully defined DEX utility functions by using incompletely defined instances of those functions. The UTADIS method performs comparatively well with functions having a high percentage of missing values.

Keywords Multi-criteria decision making · Utility function · DEX · UTADIS · Conjoint analysis · Direct marginals method

✉ Matej Mihelčić
matej.mihelcic@irb.hr

¹ Ruđer Bošković Institute, Bijenička cesta 54, 10000 Zagreb, Croatia

² Jožef Stefan International Postgraduate School, Jamova cesta 39, 1000 Ljubljana, Slovenia

³ Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia

1 Introduction

Multi criteria decision analysis (MCDA) (Figueira et al. 2005; Ishizaka and Nemery 2013) is a discipline concerned with solving decision problems that involve multiple, possibly conflicting, criteria. It provides a number of methods to create preference models by using information provided by the decision maker. This information can be given in various forms, by using different representations. Converting representations from one form to another is often highly desirable, as it can bridge the gap between different methodological approaches and enrich the capabilities of the individual ones.

A typical MCDA model consists of at least two components: a set of decision alternatives $A = \{a_1, a_2, \dots, a_m\}$ and a set of criteria $X = \{x_1, x_2, \dots, x_n\}$. The alternatives are first assessed on the individual criteria and then evaluated or ranked by some procedure, taking into account the decision maker's preferences. This procedure is often based on *preference aggregation* (Ouerdane et al. 2010), employing some type of *utility* or *value functions*. The most common form of aggregation in MCDA is based on the *weighted sum model*, where the overall value $u(a)$ of the alternative $a \in A$ is determined by the function f so that $u(a) = f(x_1(a), x_2(a), \dots, x_n(a)) = \sum_{i=1}^n w_i x_i(a)$. Here, $w_i \in \mathbb{R}$ are weights, defined by the decision maker, and $x_i(a)$ are preferences established on individual criteria. There are many other models of aggregation in MCDA (Figueira et al. 2005; Ehrgott et al. 2010); see, for instance, Greco et al. (2004) for an axiomatic characterization of general utility functions and some special cases (associative operator, Sugeno integral and ordered weighted maximum), and Yang (2001) for an overview of qualitative and quantitative aggregation under uncertainty.

This study addresses two types of utility function representations, qualitative and quantitative, and investigates how to convert the former to the latter. Specifically, we look at utility functions in the context of an MCDA method DEX. DEX (Bohanec et al. 2013) is a qualitative MCDA method, which employs discrete attributes and discrete utility functions. The latter are represented in a rule-based point-by-point way (see Sect. 2.1). This makes DEX suitable particularly for classifying decision alternatives into predefined discrete classes, i.e., solving the MCDA problem known as *sorting* (Roy 1996).

In this study, we examine and compare three methods of approximating DEX utility functions by piece-wise linear marginal utility functions: the Direct marginals method, the UTADIS method and the Conjoint analysis method. The Direct marginals method (Sect. 2.3) establishes marginal utility functions by a projection of a DEX utility function to individual attributes. UTADIS (Devaud et al. 1980; Siskos et al. 2005) (Sect. 2.4) is a quantitative method that constructs numerical additive utility functions from a provided subset of alternatives and assigns this alternatives to predefined ordered groups. The Conjoint analysis (Agarwal et al. 2015; Green and Srinivasan 1990) (Sect. 2.5) is a method that constructs numerical additive utility functions from attribute preference scores or grades described by one or more decision makers, for a given set of alternatives, by fitting a linear model to approximate the measured variable.

All three methods are aimed at providing an approximate quantitative representation of a qualitative DEX function. Why is this important? In general, the motivation for making such approximations is the same as in disaggregation–aggregation, one of the main MCDA paradigms (Siskos et al. 2005), which has produced a number of well-known MCDA methods, including UTA (Jacquet-Lagrez and Siskos 1982) and UTADIS (Devaud et al. 1980). The disaggregation–aggregation approach builds on examples of decisions, provided by the decision maker, and develops an explicit multi-criteria model that (1) represents and assesses decision maker’s preferences and (2) is capable of evaluating decision alternatives other than those originally provided by the decision maker. In the context of DEX, the approximation of utility functions extends its capabilities and is useful for several reasons. First, the newly obtained numerical evaluations facilitate an easy ranking and comparison of decision alternatives, especially those that are assigned to the same qualitative class by DEX. Consequently, the sensitivity of evaluation is increased. Second, the sheer form of numerical functions may provide additional information about the properties of underlying DEX functions, which is useful in verification, representation and justification of DEX models. The extension to incompletely defined DEX functions, which is addressed in this paper, allows these methods to be used in real world applications, which are often faced with incompleteness.

In this study, the three methods are experimentally assessed on a collection of DEX utility functions that emerged in real world decision making problems, and on several sets of artificially generated DEX utility functions of different dimensions with attributes containing different number of preference categories. We focused on the accuracy of representation of DEX utility functions containing different levels of missing values. We also study the performance of approximation methods with respect to the dimensionality of DEX utility function domains.

Previous attempts to approximate DEX utility functions include: a linear approximation method commonly used in DEX to assess criteria importance (Bohanec and Zupan 2004), an early method for ranking of alternatives and improving the sensitivity of evaluation called QQ (Mileva-Boshkoska and Bohanec 2012; Bohanec et al. 1992), DEX utility function approximation with copulas (Mileva-Boshkoska and Bohanec 2012) and by using methods UTA and ACUTA (Mihelčić and Bohanec 2014). Finally, completely defined, monotone DEX utility functions were approximated by using the Direct marginals, Conjoint analysis and UTADIS methods (Mihelčić and Bohanec 2015).

This paper builds upon our previous work on approximating monotone, complete DEX utility functions (Mihelčić and Bohanec 2015). It assesses the same three methods (Direct marginals, Conjoint analysis and UTADIS), but extends the analysis from artificially generated, completely defined, monotone DEX utility functions to (1) *incompletely defined* artificially generated monotone functions and (2) *real world* functions, extracted from DEX models that were developed in the past to support various real-life decision problems.

2 Methodology

In this section, we describe the DEX method and methods used to approximate DEX utility functions.

2.1 DEX method

DEX (Bohanec et al. 2013) is a qualitative MCDA method for the evaluation and analysis of decision alternatives, and is implemented in the software DEXi (Bohanec 2015). In DEX, all attributes are qualitative and can take values represented by words, such as “low” or “excellent”. Attributes are generally organised in a hierarchy. The evaluation of decision alternatives is carried out by utility functions, which are represented in the form of decision rules.

In the context of this paper, we focus on individual utility functions. All attributes (function arguments and outcomes) are assumed to be discrete and preferentially ordered, so that a higher ordinal value represents a better preference. In this setting, a DEX utility function f is defined over a set of attributes $\mathbf{x} = (x_1, x_2, \dots, x_n)$ so that

$$f : X_1 \times X_2 \times \dots \times X_n \rightarrow Y$$

Here, X_i , $i = 1, 2, \dots, n$, denote *value scales* of the corresponding attributes x_i , and Y is the value scale of the output attribute y . Since values of the output attribute are discrete, they are also referred to as *classes*.

In real world DEX models, values are represented verbally. A typical DEX value scale consists of a small number, usually two to five, of words. For example, it is quite common to use scales such as {unacceptable, acceptable, good, excellent} for evaluative attributes, or {high, medium, low} for attributes representing costs. In this paper, however, we will mostly assume that attribute values are represented with ordinal numbers:

$$X_i = \{1, 2, \dots, k_i\}, i = 1, 2, \dots, n \quad \text{and} \quad Y = \{1, 2, \dots, c\}$$

The function f is represented by a set of *decision rules*

$$F = \{(\mathbf{x}, y) \mid \mathbf{x} \in X_1 \times X_2 \times \dots \times X_n, y \in Y, y = f(\mathbf{x})\}$$

Each rule $(\mathbf{x}, y) \in F$ defines the mapping $\mathbf{x} \mapsto f(\mathbf{x})$ in one data point \mathbf{x} . Ideally, the set of rules covers the whole domain, so that the function is defined for all combinations of arguments' values. In this case, we say that the function is *completely defined* (or *complete*). Since all attributes are preferentially ordered, it is also expected that DEX functions are *monotone*: when argument values increase, the function value increases or remains constant.

In reality, however, these assumptions may not always hold. For practical reasons, in addition to preferentially increasing scales, the supporting DEXi software allows

the use of decreasing and unordered scales (Bohanec 2015). Some previous implementations of DEX (Bohanec et al. 2013) even did not check the ordering, so the actual ordering of attributes in some real-world functions may be unknown. It is also possible that a decision maker erroneously or deliberately (against the warning issued by the software) enters a decision rule that breaches the monotonicity. Consequently, the functions encountered in practice may not be monotone, or the actual state of monotonicity is difficult to establish.

Even more common, and somewhat expected, situations occur when the decision maker provides only a subset of rules. In this case, a utility function is *incomplete*, because it is not defined for some $\mathbf{x}^* \in X_1 \times X_2 \times \dots \times X_n$. In such cases, DEX attempts to determine the missing rule (\mathbf{x}^*, y^*) from other, already defined rules. Two methods are provided for this purpose. The first one is based on the principle of *dominance* (Greco et al. 2001): the lower and upper bounds of y^* are assessed using the assumption of monotonicity (Błaszczyszki et al. 2009). In short, to maintain the monotonicity of f , y^* should be bounded so that

$$\max_{(\mathbf{x}, y) \in F: \mathbf{x} \leq \mathbf{x}^*} y \leq y^* \leq \min_{(\mathbf{x}, y) \in F: \mathbf{x} \geq \mathbf{x}^*} y$$

The second method interpolates the defined rules by a linear function, from which the missing values y^* are determined. See Bohanec and Zupan (2004) for a detailed description.

All these practical considerations are important because they put additional requirements to methods aimed at approximating real world DEX utility functions. In general, an approximation method should expect utility functions that exhibit various degrees of incompleteness and/or violation of monotonicity. Also, it is no longer true that for each \mathbf{x} , $f(\mathbf{x})$ maps to a single value $y \in Y$. Instead, $f(\mathbf{x})$ generally maps to a subset of categories $Y_{val} \subseteq Y$, where $Y_{val} = Y \cap [y_l, y_h]$ and $y_l, y_h \in Y, y_l \leq y_h$ are determined by dominance.

Example In order to illustrate these concepts, let us use a relatively simple, but non-trivial DEX utility function of two arguments. It appears in the decision model called *Employ*, which is distributed together with the DEXi software package (Bohanec 2015) and is aimed at evaluating candidates applying for a job. The function is called *Educat*. It is positioned at the second hierarchical level of the model and aggregates two attributes $x_1 = Formal$ (candidate’s formal education) and $x_2 = For. lang$ (candidate’s mastering of foreign languages), into the assessment of $y = Educat$, candidate’s suitability for the job from the educational viewpoint. (Please note that other viewpoints, such as experience, age, and personal characteristics, are addressed in other parts of the model, but those are not relevant for the purpose of this example.) The verbal scales of the attributes are all preferentially ordered and defined as follows:

- $X_1 = X(Formal) = \{\text{prim-sec, high, univ, MSc, PhD}\}$
- $X_2 = X(For. lang) = \{\text{no, pas, act}\}$
- $Y = Y(Educat) = \{\text{unacc, acc, good}\}$

Table 1 Decision rules defining utility function *Educat*, which aggregates candidate’s *Formal* education and their mastering of *Foreign languages*

The same rules are presented verbally (left) and with ordinal values (right)

Rule	Formal	For. lang	Educat	x_1	x_2	y
1	Prim-sec	Act	Unacc	1	3	1
2	High	Pas	Unacc	2	2	1
3	High	Act	Act	2	3	2
4	Univ	Pas	Acc	3	2	2
5	Univ	Act	Acc	3	3	2
6	MSc	Act	Good	4	3	3
7	PhD	No	Unacc	5	1	1
8	PhD	Pas	Acc	5	2	2

Table 1 shows eight decision rules that were defined by the decision maker. The total number of possible decision rules $\mathbf{x} \in X_1 \times X_2$ in this case equals $|X_1| \times |X_2| = 5 \times 3 = 15$, so it is clear that this function is incompletely defined and that seven rules are missing. For instance, there is no information about $\mathbf{x}^* = (\text{MSc}, \text{pas})$. In order to determine bounds for (\mathbf{x}^*, y^*) according to the principle of dominance, all defined rules such that $\mathbf{x} \leq \mathbf{x}^*$ (i.e., rules numbered 2 and 4) are checked first. The maximum *Educat* value in these rules is “acc”, which gives the lower bound for y^* . The upper bound is established by finding the minimum evaluation in rules for which $\mathbf{x} \geq \mathbf{x}^*$ (rules 6 and 8). Again, the bound is “acc” (from rule 8). Therefore, the value of the missing rule (\mathbf{x}^*, y^*) is set to $y^* = \text{acc}$. With this function, it is actually possible, using the same procedure, to uniquely determine the values of all missing rules. The obtained utility function is graphically presented in Fig. 1.

2.2 Approximation of DEX utility functions

All methods assessed in this study are used to approximate some DEX utility function f with marginal utility functions $u_i : X_i \rightarrow \mathbb{R}, i = 1, 2, \dots, n$. The functions u_i are assumed to take a piece-wise linear form. For the alternative $\mathbf{a} = \mathbf{x}(a) = (x_1(a), x_2(a), \dots, x_n(a))$, the numeric value of $u_i(x_i(a))$ is established from f for each $x_i(a) \in X_i$, while its value for $x_i(a) \notin X_i$ is linearly interpolated from the closest neighboring points of X_i .

On this basis, f is approximated as a weighted sum of the marginal utility functions:

$$u(\mathbf{x}(a)) = u(x_1(a), x_2(a), \dots, x_n(a)) = \sum_{i=1}^n \omega_i u_i(x_i(a))$$

Here, $\omega_i \in \mathbb{R}, i = 1, 2, \dots, n$, are weights of the corresponding attributes, normalised so that $\sum_{i=1}^n \omega_i = 1$.

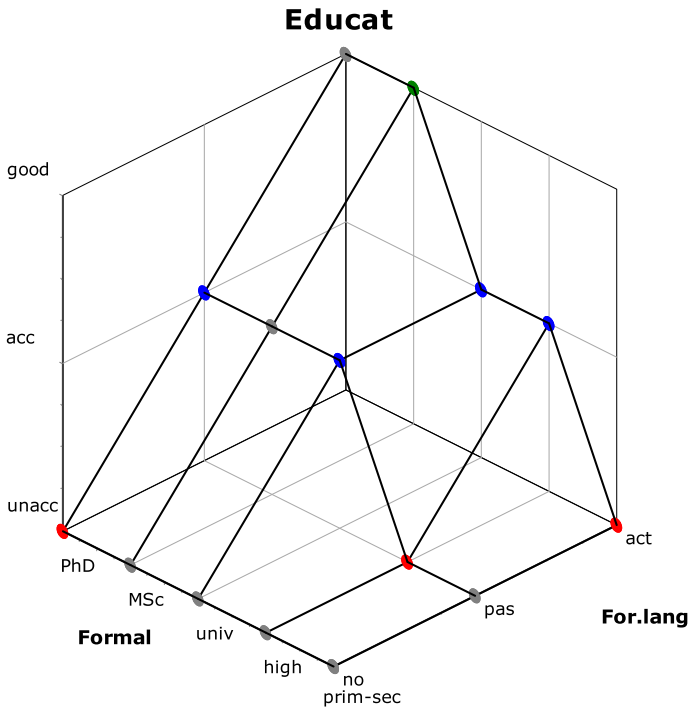


Fig. 1 Graphical representation of completely defined function *Educat*. The function is discrete and is defined only in points represented by circles; lines connecting the points serve only for visualization. The grey-colored points have been determined using the principle of dominance, and all the other points have been defined by the decision maker

2.3 Direct marginals method

The Direct marginals method (hereafter DM for short) establishes the marginal utility function $u_i(v)$ as an average value of the target attribute y for the alternative (decision rule) $a \in F$, where $x_i(a) = v$. Let $F_{i,v} \subseteq F$ denote all decision rules where $x_i(a) = v$. Then

$$u_i(v) = \frac{1}{|F_{i,v}|} \sum_{\{a \in F_{i,v}\}} y(a), \quad i = 1, 2, \dots, n, \quad v \in X_i$$

In the experiments (Sect. 3), all functions u_i were scaled to the $[0, 1]$ interval, therefore the importance weights ω_i were determined as a percentage of total utility range covered by the range of a particular attribute. For each $v \in X_i$, we compute $u_i(v)$ and define $\omega'_i = \max(u_i(v)) - \min(u_i(v))$. The importance weights are then computed as:

$$\omega_i = \frac{\omega'_i}{\sum_{j=1}^n \omega'_j}$$

DM takes advantage of the fact that DEX utility functions are defined on an orthogonal and relatively small lattice of input attributes. Usually, for each value $v \in X$, there is a number of decision rules $F_{i,v}$ that provide evidence about $u_i(v)$, which can be thus accurately assessed by averaging. This method is expected to perform well on completely defined DEX utility functions, what was confirmed in our previous study (Mihelčić and Bohanec 2015). For incompletely defined functions, the performance is expected to deteriorate; this is one of the questions investigated further in Sects. 3 and 4.

2.4 UTADIS method

The UTADIS method (Devaud et al. 1980; Siskos et al. 2005) is an extension of UTA (Utilités Additives) method (Jacquet-Lagreze and Siskos 1982). UTADIS (referred to as UD in figures and tables) enables the decision maker to assign alternatives to predefined ordered groups. Thus it is very well suited to our problem of approximating discrete DEX functions, assuming that each DEX decision rule $a \in F$ defines some (hypothetical) decision alternative.

For each attribute x_i , let x_{i_l} and x_{i_h} represent the most and the least preferred value of the attribute. Each attribute range $[x_{i_l}, x_{i_h}]$ is divided into $k_i - 1$ subintervals $[x_i^\alpha, x_i^{\alpha+1}]$, $\alpha = 1, 2, \dots, k_i - 1$. UTADIS approximates the marginal utility function u_i as:

$$u_i(x_i(a)) = u_i(x_i^J) + \frac{x_i(a) - x_i^J}{x_i^{J+1} - x_i^J} [u_i(x_i^{J+1}) - u_i(x_i^J)], \quad 1 \leq J \leq k_i - 1$$

The value J is chosen so that $x_i(a) \in [x_i^J, x_i^{J+1}]$.

The alternatives are assigned to groups by using thresholds t_i : $u(x(a)) \geq t_1 \Rightarrow a \in C_1$, $t_2 \leq u(x(a)) < t_1 \Rightarrow a \in C_2, \dots, u(x(a)) < t_{c-1} \Rightarrow a \in C_c$, where $u(x(a)) = \sum_{i=1}^n u_i(x_i(a))$.

UTADIS searches for marginal utility functions by solving the linear programming problem $\min E = \sum_{k=1}^c \frac{\sum_{a_j \in C_k} \sigma(a)_j^+ + \sigma(a)_j^-}{m_k}$, where σ^+, σ^- denote errors after violation of lower/upper bound of a group C_k , and m_k denotes the number of alternatives assigned to the group C_k .

The constraints of the linear programming problem are:

$$\begin{aligned} & \sum_{i=1}^n \left(u_i(x_i^J) + \frac{x_i(a) - x_i^J}{x_i^{J+1} - x_i^J} [u_i(x_i^{J+1}) - u_i(x_i^J)] \right) - u_1 + \sigma_j^+ \geq \delta_1, \quad \forall a \in C_1, \\ & \sum_{i=1}^n \left(u_i(x_i^J) + \frac{x_i(a) - x_i^J}{x_i^{J+1} - x_i^J} [u_i(x_i^{J+1}) - u_i(x_i^J)] \right) - u_k + \sigma_j^+ \geq \delta_1, \\ & \sum_{i=1}^n \left(u_i(x_i^J) + \frac{x_i(a) - x_i^J}{x_i^{J+1} - x_i^J} [u_i(x_i^{J+1}) - u_i(x_i^J)] \right) - u_{k-1} - \sigma_j^- \leq -\delta_2, \end{aligned}$$

$$\begin{aligned} &\forall a \in C_k, (k = 2, 3, \dots, c - 1), \\ &\sum_{i=1}^n \left(u_i(x_i^J) + \frac{x_i(a) - x_i^J}{x_i^{J+1} - x_i^J} [u_i(x_i^{J+1}) - u_i(x_i^J)] \right) - u_{c-1} - \sigma_j^- \leq -\delta_2, \forall a \in C_c, \\ &\sum_{i=1}^n \sum_{j=1}^{k_i-1} [u_i(x_i^{j+1}) - u_i(x_i^j)] = 1, \\ &u_k - u_{k+1} \geq s, \forall k = 1, 2, \dots, c - 2, \\ &(u_i(x_i^{j+1}) - u_i(x_i^j)) \geq 0, \sigma_j^+ \geq 0, \sigma_j^- \geq 0, \forall i = 1, 2, \dots, n, \end{aligned}$$

Here, δ_1, δ_2 and s are small positive constants such that $s > \delta_1, \delta_2 \geq 0$. After the optimal solution is found, a post optimality stage is carried out to access the stability of obtained solution [Devaud et al. \(1980\)](#).

2.5 Conjoint analysis method

The Conjoint analysis (CA) method ([Agarwal et al. 2015](#); [Green and Srinivasan 1990](#)) is aimed at explaining decision maker’s preferences. It determines the importances of attributes, their interactions and utility functions for each attribute in a decision making problem. In general, it can take into account preferences from many different decision makers and use them to asses attribute importance and the values of appropriate attribute levels. There are several variations of CA. In this work, we used the one implemented in the R package ‘Conjoint’ ([Bak 2012](#)).

The input to the method is a DEXi table F . The method creates a linear fit of the data $Y = \alpha + X\beta + \varepsilon$, where $Y \in \mathbb{R}^n, \alpha \in \mathbb{R}, X \in \mathbb{R}^{n \times p}, \beta \in \mathbb{R}^p, \varepsilon \sim N(0, \sigma^2 I)$. The procedure takes as input a matrix $X_0 = [\mathbb{1} \ X]$ and $\beta_0 = [\alpha \ \beta]$. A linear model is found by performing a QR decomposition of a matrix $X_0, (X_0 = QR = Q \begin{bmatrix} R_1 \\ 0 \end{bmatrix})$. The solution is obtained by solving $Q^T Y = R\beta$, and the parameters are estimated by using the least squares method:

$$\min_{\hat{s}} \|R\beta \cdot \hat{s} - Q^T Y\|_2^2$$

For the singular input matrices, the least squares method does not provide a unique solution. In the implementation that we use, a column is discarded from the matrix if its diagonal element in R_β is 10^{-7} times smaller than the largest diagonal value. Thus, this implementation may produce an undefined value for a subset of coefficients when approximating some incomplete DEX input functions. When this happens, we consider that CA did not succeed to approximate the given function.

3 Experimental set up

In this section, we describe the experimental set up and explain the procedures used to perform our experiments.

The goal of the experiments was to assess and compare the performance of the three methods – DM, UTADIS, and CA – on artificially generated, monotone, possibly incomplete DEX utility functions (Sect. 3.1) and on a set of functions that were constructed in real-world decision problems (Sect. 3.2).

All experiments were performed in the R programming language by using the 'MCDA' (Meyer and Bigare 2015), the 'Conjoint' (Bak 2012) and the 'pROC' (Robin et al. 2011) R packages. In addition, we implemented the DM method, the RMSE measure, a monotone function generator that generates all monotone functions in some space with given dimensions, a random monotone function generator that generates a defined number of random monotone functions from a space with given dimensions, the procedure for introducing missing values and a procedure for determining the ordering of attributes in incompletely defined functions. The ordering procedure is used to determine whether a given attribute is preferentially ordered or not, and if so, in which direction, increasing or decreasing. This information is needed by UTADIS in order to properly set up the optimization constraints.

3.1 Approximating artificial DEX utility functions

For the experiments involving artificial utility functions, we generated all monotone functions for domains with dimensions $3 \times 3 \rightarrow 4$, $3 \times 4 \rightarrow 3$, $4 \times 4 \rightarrow 3$ and $5 \times 6 \rightarrow 7$ (the notation $3 \times 3 \rightarrow 4$ denotes the set of all monotone utility functions having two three-valued arguments that map to 4 values). For larger domains, the evaluation was performed on several randomly generated function sets of different sizes: $3 \times 4 \times 3 \times 5 \rightarrow 6$, $4 \times 5 \times 5 \rightarrow 6$, $5 \times 6 \rightarrow 7$, $6 \times 7 \rightarrow 7$, $8 \times 7 \rightarrow 7$ containing 1000 monotone functions, and $3 \times 5 \times 3 \times 4 \rightarrow 4$ containing 100 monotone functions. For each set of functions, we assessed the performance of methods for completely defined functions and functions containing 10, 20 % and 50 % missing values. On these sets, the quality of approximations was evaluated only on the defined points of the incompletely defined functions. To introduce missing values, a predefined proportion of decision rules was randomly removed from each DEX table. We used a random sample containing 1000 functions of dimensions $4 \times 4 \rightarrow 3$, $3 \times 4 \rightarrow 4$, $5 \times 6 \rightarrow 7$ and $6 \times 7 \rightarrow 7$ to assess the methods' ability to approximate a completely defined utility function from the partially defined function.

3.2 Approximating real world DEX utility functions

A similar procedure was also applied on real world DEX utility functions. For this purpose, we created a dataset of 6362 utility functions, extracted from an archive of 582 DEX models. These models were developed in the past in various circumstances, from large-scale international projects to students' assignments (see Bohanec et al. (2013) for an overview of DEX applications). All these models are "real" in the sense that they were developed by real people with specific decision problems in mind. The functions are of very different sizes, from $2 \rightarrow 2$ (one argument only) to $3 \times 3 \times 3 \times 3 \times 3 \times 3 \times 3 \rightarrow 5$ (seven arguments) and $2 \times 2 \times 2 \times 3 \times 2 \times 2 \times 2 \times 3 \rightarrow 5$ (eight arguments). In average, they have 2.5 arguments and map to 3.7 classes. The average

and median of domain size are 39.3 and 16, respectively. All possible rules were developed in 79.13 % of functions, but, due to the use of methods for determining missing rules' values, as much as 92.57 % of functions are complete. 93.25 % of functions are monotone. Among the 6362 functions, 3062 functions are unique in the sense that they differ from all other unique functions in at least one decision rule. Experiments were performed only on unique functions, but the results were weighted with the frequency of their occurrence in the original dataset.

3.3 Experimental procedure and evaluation measures

The experimental procedure consists of approximating DEX utility functions by using the three methods, and computing evaluation scores for each method's resulting approximated utility function.

Two measures were used for the evaluation: the Area Under the Curve (AUC) (Fawcett 2006) and the Root Mean Squared Error (RMSE) (Hyndman and Koehler 2006). To compute the AUC measure, we scaled the utility scores for the target attributes computed by the methods to the $[0, 1]$ interval. These scores were used along with the corresponding target values to compute AUC. For the RMSE measure, all the utility scores were scaled to the $[y_l, y_h]$ interval and the RMSE was computed from the difference between the approximate values obtained by the methods and the real target class value. Finally, the average and the standard deviation of AUC and RMSE were computed for sets of functions with given dimensions, to assess and compare methods' performance on whole function sets.

We only report performance for the subset of functions for which all methods returned a valid solution, however we provide the information on the number of successes for each method.

The quality of approximation of real world DEX utility functions, which generally contain missing values, were evaluated only on defined points of functions, since the completely defined functions were unknown. With the artificially created utility functions, we began with completely defined functions and then removed decision rules to introduce incompleteness. Thus, the completely defined functions were known, which allowed a dual evaluation: on all or only defined points of each incomplete function. The defined points of a function were determined by the rules entered by the user or uniquely determined by the dominance relation.

The experiments to assess approximation accuracy on all points of fully defined function were performed as follows:

- Generate 50 random sample copies of a given function, reduced so as to contain $k\%$ of missing values.
- Approximate each incomplete function with all three methods.
- For each incomplete function, assess the approximation accuracy on a completely defined rule set and take the average value as a final score.
- Compute the average and standard deviation of the measures across all generated functions.

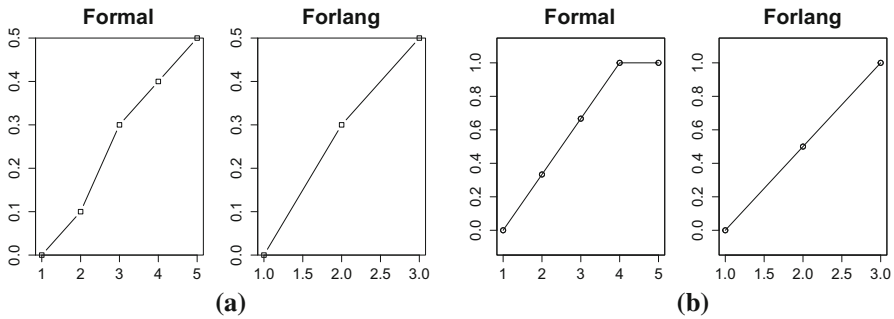


Fig. 2 Approximation of a fully defined *Educat* function produced by **a** UTADIS: AUC = 1, RMSE = 0.580 on a fully defined function. $\omega_1 = \omega_2 = 0.5$ and **b** DM and CA: AUC = 1, RMSE = 0.600 on a fully defined function. $\omega_1 = 0.45, \omega_2 = 0.55$

The repeated sample generation was performed to obtain stable results, which appear to be very sensitive to the subset of decision rules that are removed from the original function definition.

4 Results

In this section, we analyse the results of approximation of various DEX utility functions.

4.1 Results on *Educat* function

First, let us illustrate the performance of methods on the single function *Educat*, as defined in Table 1 and Fig. 1. On a completely defined function, UTADIS created two equally-weighted marginal utility functions shown in Fig. 2a, with AUC = 1 and RMSE = 0.58. Both DM and CA constructed exactly the same marginal utility functions, shown in Fig. 2b, with AUC = 1 and RMSE = 0.60. The weights ω_1 and ω_2 , associated with the marginal functions, are 0.45 and 0.55, respectively.

Overall, the methods performed equally in terms of AUC, however, in terms of RMSE, UTADIS performed slightly better. Nevertheless, it seems that DM and CA better captured the decision maker's preferences regarding the *Formal* education of the candidate. Notice that Fig. 1 shows exactly the same function values for the *Formal* education levels of MSc and PhD, indicating decision maker's indifference between these two levels. This indifference was properly captured by DM and CA (Fig. 2b), but not by UTADIS (Fig. 2a), where the utility of value *Formal*=5 (PhD) was largely overestimated.

The methods' performance changes when decision rules are removed and the function definition becomes incomplete. Fig. 3a–c show the constructed marginal utility functions of the three methods, respectively, with 53 % of decision rules missing. The attribute importance for DM is 0.67 and 0.33, and for CA 0.57 and 0.43. In this case, UTADIS' results remain exactly the same, while the results of the other two methods

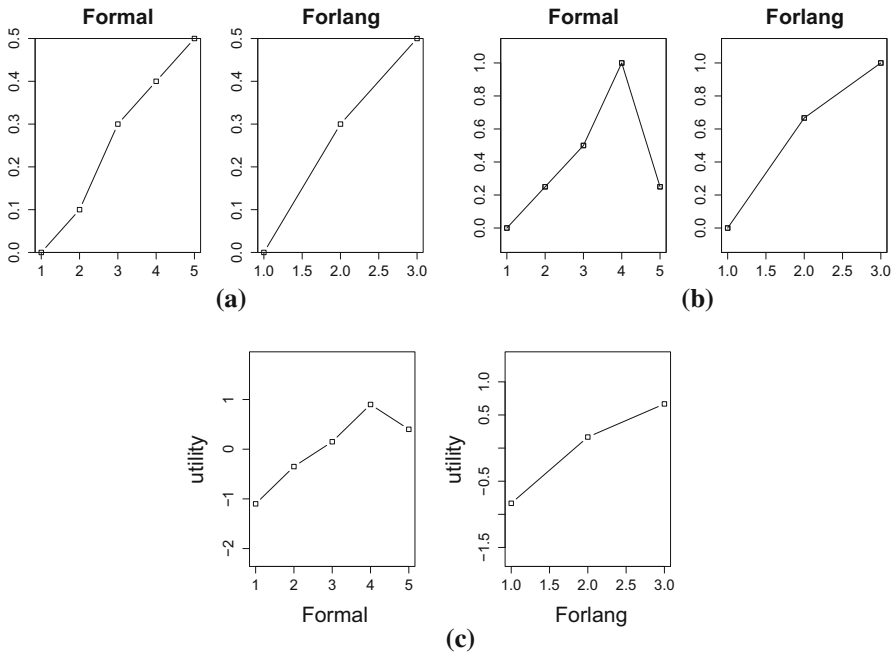


Fig. 3 Approximation of the *Educat* function containing 53 % missing values produced by **a** UTADIS: AUC = 1, RMSE = 0.660 on the incomplete function and AUC = 1, RMSE = 0.580 on the complete one. $\omega_1 = \omega_2 = 0.5$, **b** DM: AUC = 0.986, RMSE = 0.306 on the incomplete function and AUC = 0.829, RMSE = 0.604 on the complete one. $\omega_1 = 0.67, \omega_2 = 0.33$ and **c** CA: AUC = 1, RMSE = 0.177 on the incomplete function and AUC = 0.983, RMSE = 0.636 on the complete one. $\omega_1 = 0.57, \omega_2 = 0.43$

deteriorate considerably. In Fig. 3b, c, the indifference between MSc and PhD is no longer properly identified and, furthermore, the marginal utility functions of *Formal* are not monotone any more, breaching the principle of dominance.

In summary, this illustrative example indicates a similar performance of the methods on a completely defined function, and a better, more stable, performance of UTADIS on an incompletely defined function.

4.2 Results on artificial DEX functions

The results in Tables 2, 3 and 4 show that the methods produce highly accurate approximations on completely defined functions. The average AUC is generally high, in most cases is close to 1, and falls below 0.9 only in very large domains, which are rarely encountered in practice. The average RMSE also indicates good performance; in small domains with 3 or 4 classes, it is around or below 0.5. It gradually increases with the number of classes, and typically exceeds 1 only in very large domains. The approximations are also fairly accurate on incomplete functions when assessed with respect to the defined points of the function. In some cases, especially for CA, the performance even improves when evaluated on functions with 50 % missing values.

Table 2 Comparison results for DM, CA and UTADIS on various generated DEX monotone utility functions

Meth.	Space dimension	Num.	Avg. AUC	Avg. RMSE	Succ. (%)	mv. (%)
DM	$3 \times 3 \rightarrow 4$	980	0.996 ± 0.015	0.528 ± 0.239	99.2	0
	$3 \times 3 \rightarrow 4$	980	0.988 ± 0.029	0.544 ± 0.230	98.5	10
	$3 \times 3 \rightarrow 4$	980	0.987 ± 0.032	0.539 ± 0.227	97.6	20
	$3 \times 3 \rightarrow 4$	980	0.979 ± 0.044	0.517 ± 0.255	79.7	50
	$3 \times 4 \rightarrow 3$	490	0.998 ± 0.011	0.404 ± 0.162	99.4	0
	$3 \times 4 \rightarrow 3$	490	0.995 ± 0.018	0.406 ± 0.146	99.2	10
	$3 \times 4 \rightarrow 3$	490	0.992 ± 0.023	0.400 ± 0.145	98.8	20
	$3 \times 4 \rightarrow 3$	490	0.989 ± 0.032	0.364 ± 0.171	68.6	50
	$4 \times 4 \rightarrow 3$	1764	0.995 ± 0.013	0.416 ± 0.135	99.8	0
	$4 \times 4 \rightarrow 3$	1764	0.992 ± 0.019	0.412 ± 0.124	99.7	10
	$4 \times 4 \rightarrow 3$	1764	0.988 ± 0.026	0.416 ± 0.128	99.6	20
	$4 \times 4 \rightarrow 3$	1764	0.987 ± 0.031	0.382 ± 0.146	96	50
	$5 \times 6 \rightarrow 7$	1000	0.983 ± 0.020	0.970 ± 0.356	100	0
	$5 \times 6 \rightarrow 7$	1000	0.982 ± 0.020	0.975 ± 0.369	100	10
	$5 \times 6 \rightarrow 7$	1000	0.983 ± 0.019	0.990 ± 0.356	100	20
	$5 \times 6 \rightarrow 7$	1000	0.982 ± 0.020	0.984 ± 0.339	99.8	50
CA	$3 \times 3 \rightarrow 4$	980	0.997 ± 0.012	0.528 ± 0.239	99.2	0
	$3 \times 3 \rightarrow 4$	980	0.998 ± 0.029	0.501 ± 0.253	99.5	10
	$3 \times 3 \rightarrow 4$	980	0.999 ± 0.007	0.441 ± 0.272	97.6	20
	$3 \times 3 \rightarrow 4$	980	1.0 ± 0.0	0.225 ± 0.320	75.2	50
	$3 \times 4 \rightarrow 3$	490	0.998 ± 0.011	0.404 ± 0.162	99.4	0
	$3 \times 4 \rightarrow 3$	490	0.998 ± 0.011	0.389 ± 0.167	99.2	10
	$3 \times 4 \rightarrow 3$	490	0.999 ± 0.007	0.364 ± 0.176	98.8	20
	$3 \times 4 \rightarrow 3$	490	1.0 ± 0.0	0.163 ± 0.240	62.2	50
	$4 \times 4 \rightarrow 3$	1764	0.995 ± 0.013	0.416 ± 0.135	99.8	0
	$4 \times 4 \rightarrow 3$	1764	0.997 ± 0.011	0.393 ± 0.142	99.7	10
	$4 \times 4 \rightarrow 3$	1764	0.997 ± 0.013	0.385 ± 0.154	99.6	20
	$4 \times 4 \rightarrow 3$	1764	1.0 ± 0.003	0.215 ± 0.215	91.2	50
	$5 \times 6 \rightarrow 7$	1000	0.983 ± 0.020	0.970 ± 0.356	100	0
	$5 \times 6 \rightarrow 7$	1000	0.982 ± 0.020	0.958 ± 0.369	100	10
	$5 \times 6 \rightarrow 7$	1000	0.984 ± 0.019	0.931 ± 0.374	100	20
	$5 \times 6 \rightarrow 7$	1000	0.992 ± 0.017	0.730 ± 0.426	98.7	50
UD	$3 \times 3 \rightarrow 4$	980	0.970 ± 0.063	0.721 ± 0.293	99.2	0
	$3 \times 3 \rightarrow 4$	980	0.965 ± 0.075	0.748 ± 0.312	98.5	10
	$3 \times 3 \rightarrow 4$	980	0.968 ± 0.074	0.761 ± 0.321	97.6	20
	$3 \times 3 \rightarrow 4$	980	0.965 ± 0.089	0.791 ± 0.386	79.7	50
	$3 \times 4 \rightarrow 3$	490	0.976 ± 0.064	0.567 ± 0.215	99.4	0
	$3 \times 4 \rightarrow 3$	490	0.972 ± 0.067	0.576 ± 0.220	99.2	10
	$3 \times 4 \rightarrow 3$	490	0.970 ± 0.080	0.576 ± 0.229	98.8	20

Table 2 continued

Meth.	Space dimension	Num.	Avg. AUC	Avg. RMSE	Succ. (%)	mv. (%)
	$3 \times 4 \rightarrow 3$	490	0.965 ± 0.086	0.619 ± 0.265	68.6	50
	$4 \times 4 \rightarrow 3$	1764	0.972 ± 0.069	0.567 ± 0.212	99.8	0
	$4 \times 4 \rightarrow 3$	1764	0.971 ± 0.069	0.579 ± 0.223	99.7	10
	$4 \times 4 \rightarrow 3$	1764	0.968 ± 0.074	0.592 ± 0.224	99.6	20
	$4 \times 4 \rightarrow 3$	1764	0.958 ± 0.105	0.646 ± 0.272	96	50
	$5 \times 6 \rightarrow 7$	1000	0.934 ± 0.065	1.579 ± 0.712	100	0
	$5 \times 6 \rightarrow 7$	1000	0.931 ± 0.068	1.554 ± 0.694	100	10
	$5 \times 6 \rightarrow 7$	1000	0.937 ± 0.067	1.586 ± 0.716	100	20
	$5 \times 6 \rightarrow 7$	1000	0.936 ± 0.086	1.702 ± 0.832	99.6	50

For each method, space dimensions and amount of missing values (mv.), the columns show the number of utility functions (num.), average AUC and RMSE with standard deviation, and the percentage of successfully approximated functions (succ.). All the results are computed on the defined points of the functions

Table 3 Comparison results for DM, CA and UTADIS on various generated DEX monotone utility functions

Meth.	Space dimension	Num.	Avg. AUC	Avg. RMSE	Succ. (%)	mv. (%)
DM	$6 \times 7 \rightarrow 7$	1000	0.979 ± 0.019	0.974 ± 0.339	100	0
	$6 \times 7 \rightarrow 7$	1000	0.979 ± 0.020	0.990 ± 0.335	100	10
	$6 \times 7 \rightarrow 7$	1000	0.978 ± 0.021	1.010 ± 0.316	100	20
	$6 \times 7 \rightarrow 7$	1000	0.979 ± 0.022	0.989 ± 0.338	100	50
	$8 \times 7 \rightarrow 7$	1000	0.975 ± 0.022	1.001 ± 0.318	100	0
	$8 \times 7 \rightarrow 7$	1000	0.977 ± 0.022	0.986 ± 0.315	100	10
	$8 \times 7 \rightarrow 7$	1000	0.975 ± 0.022	0.994 ± 0.321	100	20
	$8 \times 7 \rightarrow 7$	1000	0.975 ± 0.022	0.980 ± 0.308	99.9	50
	$4 \times 5 \times 5 \rightarrow 6$	1000	0.944 ± 0.026	1.056 ± 0.248	100	0
	$4 \times 5 \times 5 \rightarrow 6$	1000	0.945 ± 0.026	1.037 ± 0.247	100	10
	$4 \times 5 \times 5 \rightarrow 6$	1000	0.945 ± 0.026	1.033 ± 0.239	100	20
	$4 \times 5 \times 5 \rightarrow 6$	1000	0.945 ± 0.027	1.039 ± 0.247	100	50
	$3 \times 4 \times 3 \times 5 \rightarrow 6$	1000	0.923 ± 0.026	1.150 ± 0.220	100	0
	$3 \times 4 \times 3 \times 5 \rightarrow 6$	1000	0.922 ± 0.027	1.146 ± 0.216	100	10
	$3 \times 4 \times 3 \times 5 \rightarrow 6$	1000	0.921 ± 0.026	1.149 ± 0.226	100	20
	$3 \times 4 \times 3 \times 5 \rightarrow 6$	1000	0.920 ± 0.027	1.161 ± 0.220	100	50
	$6 \times 7 \rightarrow 7$	1000	0.979 ± 0.019	0.974 ± 0.339	100	0
	$6 \times 7 \rightarrow 7$	1000	0.980 ± 0.020	0.972 ± 0.341	100	10
	$6 \times 7 \rightarrow 7$	1000	0.978 ± 0.022	0.966 ± 0.341	100	20
	$6 \times 7 \rightarrow 7$	1000	0.987 ± 0.019	0.787 ± 0.404	99.7	50

Table 3 continued

Meth.	Space dimension	Num.	Avg. AUC	Avg. RMSE	Succ. (%)	mv. (%)
CA	$8 \times 7 \rightarrow 7$	1000	0.975 ± 0.022	1.001 ± 0.318	100	0
	$8 \times 7 \rightarrow 7$	1000	0.976 ± 0.021	0.982 ± 0.332	100	10
	$8 \times 7 \rightarrow 7$	1000	0.975 ± 0.022	0.967 ± 0.335	100	20
	$8 \times 7 \rightarrow 7$	1000	0.983 ± 0.020	0.821 ± 0.361	99.8	50
	$4 \times 5 \times 5 \rightarrow 6$	1000	0.944 ± 0.026	1.056 ± 0.248	100	0
	$4 \times 5 \times 5 \rightarrow 6$	1000	0.946 ± 0.026	1.033 ± 0.252	100	10
	$4 \times 5 \times 5 \rightarrow 6$	1000	0.946 ± 0.026	1.022 ± 0.248	100	20
	$4 \times 5 \times 5 \rightarrow 6$	1000	0.953 ± 0.027	0.982 ± 0.271	100	50
	$3 \times 4 \times 3 \times 5 \rightarrow 6$	1000	0.923 ± 0.026	1.150 ± 0.220	100	0
	$3 \times 4 \times 3 \times 5 \rightarrow 6$	1000	0.923 ± 0.026	1.141 ± 0.218	100	10
	$3 \times 4 \times 3 \times 5 \rightarrow 6$	1000	0.923 ± 0.026	1.139 ± 0.232	100	20
	$3 \times 4 \times 3 \times 5 \rightarrow 6$	1000	0.925 ± 0.028	1.115 ± 0.239	100	50
	UD	$6 \times 7 \rightarrow 7$	1000	0.926 ± 0.063	1.549 ± 0.642	100
$6 \times 7 \rightarrow 7$		1000	0.926 ± 0.068	1.563 ± 0.644	100	10
$6 \times 7 \rightarrow 7$		1000	0.930 ± 0.065	1.587 ± 0.685	100	20
$6 \times 7 \rightarrow 7$		1000	0.927 ± 0.085	1.687 ± 0.805	99.8	50
$8 \times 7 \rightarrow 7$		1000	0.922 ± 0.060	1.517 ± 0.600	100	0
$8 \times 7 \rightarrow 7$		1000	0.912 ± 0.067	1.573 ± 0.658	100	10
$8 \times 7 \rightarrow 7$		1000	0.920 ± 0.067	1.552 ± 0.661	100	20
$8 \times 7 \rightarrow 7$		1000	0.924 ± 0.079	1.622 ± 0.742	99.8	50
$4 \times 5 \times 5 \rightarrow 6$		1000	0.893 ± 0.055	1.311 ± 0.393	100	0
$4 \times 5 \times 5 \rightarrow 6$		1000	0.893 ± 0.057	1.301 ± 0.409	100	10
$4 \times 5 \times 5 \rightarrow 6$		1000	0.893 ± 0.058	1.301 ± 0.400	100	20
$4 \times 5 \times 5 \rightarrow 6$		1000	0.894 ± 0.065	1.356 ± 0.429	100	50
$3 \times 4 \times 3 \times 5 \rightarrow 6$		1000	0.886 ± 0.047	1.289 ± 0.315	100	0
$3 \times 4 \times 3 \times 5 \rightarrow 6$		1000	0.883 ± 0.048	1.284 ± 0.292	100	10
$3 \times 4 \times 3 \times 5 \rightarrow 6$		1000	0.880 ± 0.049	1.299 ± 0.315	100	20
$3 \times 4 \times 3 \times 5 \rightarrow 6$		1000	0.878 ± 0.050	1.330 ± 0.323	100	50

For each method, space dimensions and amount of missing values (mv.), the columns show the number of utility functions (num.), average AUC and RMSE with standard deviation, and the percentage of successfully approximated functions (succ.). All the results are computed on the defined points of the functions

In general, CA has the best performance, but is closely followed by DM. UTADIS is outperformed according to AUC and RMSE on all sets of functions.

However, it is more informative to assess how good are these approximations with respect to the completely defined functions. The results are presented in full in Table 5, and are partly visualized in Figs. 4a, b, 5a, b. Figure 4a, b show AUC and RMSE, respectively, for some selected dimensions and incompleteness levels 0 and 50%. At 0%, DM and CA perform exactly the same, while the average performance of UTADIS is somewhat worse (lower AUC and higher RMSE). At the 50%

Table 4 Comparison results for DM, CA and UTADIS on various generated DEX monotone utility functions

Meth.	Space dimension	Num.	Avg. AUC	Avg. RMSE	Succ. (%)	mv. (%)
DM	$3 \times 4 \times 5 \times 3 \times 4 \rightarrow 4$	100	0.926 ± 0.018	0.819 ± 0.090	100	0
	$3 \times 4 \times 5 \times 3 \times 4 \rightarrow 4$	100	0.926 ± 0.022	0.819 ± 0.111	100	10
	$3 \times 4 \times 5 \times 3 \times 4 \rightarrow 4$	100	0.927 ± 0.016	0.814 ± 0.104	100	20
	$3 \times 4 \times 5 \times 3 \times 4 \rightarrow 4$	100	0.927 ± 0.018	0.811 ± 0.124	100	50
CA	$3 \times 4 \times 5 \times 3 \times 4 \rightarrow 4$	100	0.926 ± 0.018	0.819 ± 0.090	100	0
	$3 \times 4 \times 5 \times 3 \times 4 \rightarrow 4$	100	0.926 ± 0.023	0.818 ± 0.112	100	10
	$3 \times 4 \times 5 \times 3 \times 4 \rightarrow 4$	100	0.928 ± 0.017	0.809 ± 0.107	100	20
	$3 \times 4 \times 5 \times 3 \times 4 \rightarrow 4$	100	0.928 ± 0.019	0.796 ± 0.126	100	50
UD	$3 \times 4 \times 5 \times 3 \times 4 \rightarrow 4$	100	0.914 ± 0.030	0.873 ± 0.152	100	0
	$3 \times 4 \times 5 \times 3 \times 4 \rightarrow 4$	100	0.907 ± 0.038	0.870 ± 0.149	100	10
	$3 \times 4 \times 5 \times 3 \times 4 \rightarrow 4$	100	0.913 ± 0.030	0.844 ± 0.114	100	20
	$3 \times 4 \times 5 \times 3 \times 4 \rightarrow 4$	100	0.906 ± 0.037	0.854 ± 0.139	100	50

For each method, space dimensions and amount of missing values (mv.), the columns show the number of utility functions (num.), average AUC and RMSE with standard deviation, and the percentage of successfully approximated functions (succ.). All the results are computed on the defined points of the functions

incompleteness level, the performance of DM is impaired the most, while the other two methods perform consistently, but only moderately worse.

Overall, the results indicate a consistent decrease of approximation accuracy with the introduction of missing values for all methods. CA has the best accuracy on this set, and DM is second best when functions contain 10 or 20 % missing values. However, it is outperformed by UTADIS on functions that contain 50 % missing values with respect to AUC.

Figure 5a, b display comparative distributions of AUC and RMSE, respectively, for a sample containing 1000 randomly generated $5 \times 6 \rightarrow 7$ functions. For each generated function, 50 random rule samplings were performed and RMSE, AUC measures were averaged on the corresponding incompletely defined functions. The results indicate a greater variability of the results of UTADIS than DM and CA. There is a number of outlier functions present for each method on which these methods achieve much lower accuracy than on the majority of other functions of the same dimensions.

Based on these findings, an interesting future work direction would be to understand if there exists a fixed set of functions that are hard to approximate for all presented methods, or is there a distinct set of “hard” functions for each method. In the latter case, it might help to use a combined approach to improve the accuracy and gain confidence in the approximations.

4.3 Results on real world DEX functions

The approximations on real world DEX functions were evaluated only on the defined points of functions because completely defined functions were generally unknown.

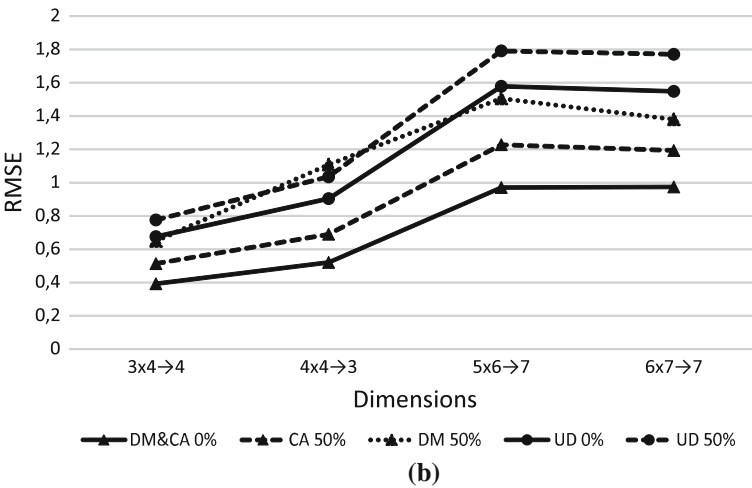
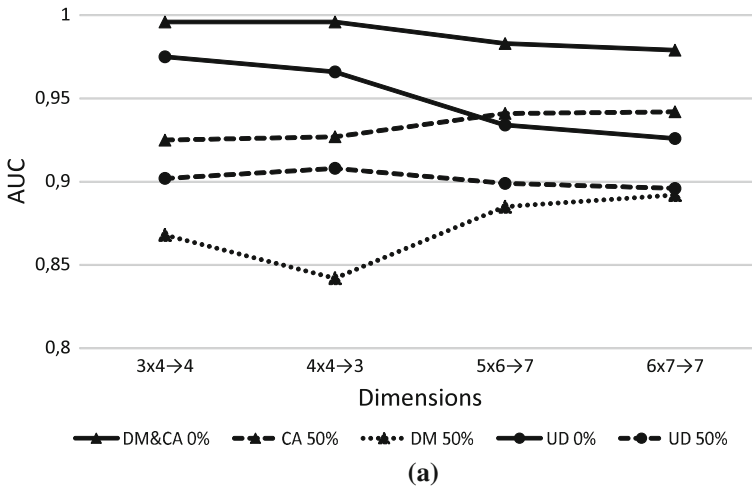


Fig. 4 Graphical representation of AUC and RMSE from Table 5 for some selected dimensions and incompleteness levels 0 and 50%. At 0%, DM and CA perform exactly the same

The results of approximations for all methods are presented in Table 6. They indicate that the methods successfully approximated about 76 % of available real world functions. The reason for such a high number of unsuccessful executions is that we wanted to make a fair comparisons of all three methods. Since UTADIS requires information about optimization direction for each attribute, we eliminated all DEX utility functions for which this direction could not had been accurately determined from available function definitions. This eliminated 719 out of 3062 (23.5 %) unique functions. From the remaining 2343 functions, DM and UTADIS methods successfully approximated all functions and CA successfully approximated 99.19 % of functions.

All the methods produced very accurate approximations when evaluated on the defined points of the functions, but as explained in Sect. 3.2, as much as 92.57 % of

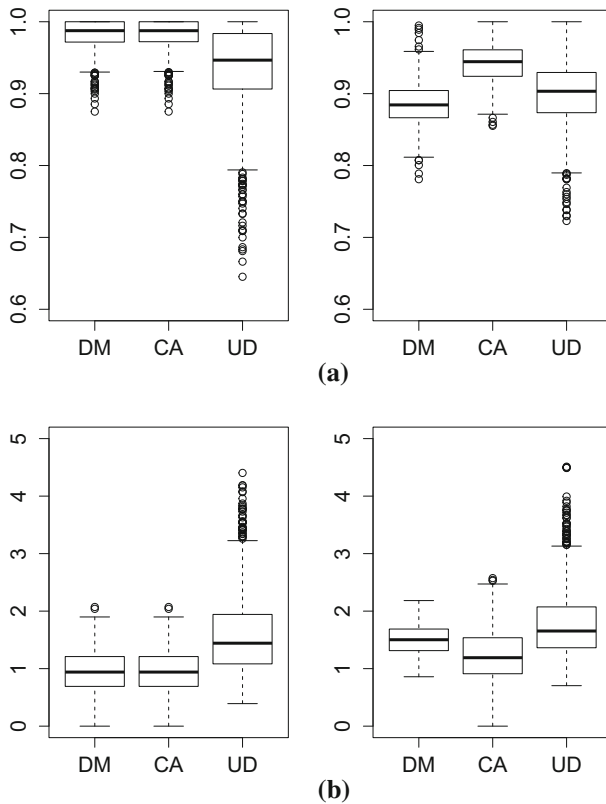


Fig. 5 Distribution of the average **a** AUC and **b** RMSE for functions containing 0% (*left*) and 50% (*right*) missing values. **a** Distribution of average AUC obtained by DM, CA and UTADIS. The performance is measured on a sample of 1000 randomly generated functions of dimensions $5 \times 6 \rightarrow 7$. Each function contains 0% (*left*) and 50% (*right*) missing values. **b** Distribution of average RMSE obtained by DM, CA and UTADIS. The performance is measured on a sample of 1000 randomly generated functions of dimensions $5 \times 6 \rightarrow 7$. Each function contains 0% (*left*) and 50% (*right*) missing values

functions were complete, so this was somewhat expected from the results achieved on artificial functions. Among the three methods, CA appears the best by a small margin; in the cases when it fails to produce a result, it can be safely replaced by DM.

5 Conclusion and future work

In this work we used three different quantitative MCDA methods to approximate utility functions of a qualitative MCDA method DEX. The methods produce visual representations in the form of continuous, piecewise linear marginal utility functions, which help in analysing and understanding the given DEX utility functions (originally represented as tables of decision rules). From obtained marginal functions, the users can see, for each attribute, the preference direction between different categories, importance levels of each category, and get the information about the overall

Table 5 Comparison results for DM, CA and UTADIS method on various generated DEX monotone utility functions

Meth.	Space dimension	Num.	Avg. AUC	Avg. RMSE	Succ. (%)	mv. (%)
DM	$4 \times 4 \rightarrow 3$	1000	0.996 ± 0.013	0.521 ± 0.249	100	0
	$4 \times 4 \rightarrow 3$	1000	0.983 ± 0.017	0.577 ± 0.205	100	10
	$4 \times 4 \rightarrow 3$	1000	0.969 ± 0.022	0.620 ± 0.189	100	20
	$4 \times 4 \rightarrow 3$	1000	0.842 ± 0.047	1.108 ± 0.141	100	50
	$3 \times 4 \rightarrow 4$	1000	0.996 ± 0.013	0.393 ± 0.172	100	0
	$3 \times 4 \rightarrow 4$	1000	0.982 ± 0.018	0.435 ± 0.127	100	10
	$3 \times 4 \rightarrow 4$	1000	0.971 ± 0.022	0.457 ± 0.116	100	20
	$3 \times 4 \rightarrow 4$	1000	0.868 ± 0.045	0.650 ± 0.084	100	50
	$5 \times 6 \rightarrow 7$	1000	0.983 ± 0.020	0.970 ± 0.356	100	0
	$5 \times 6 \rightarrow 7$	1000	0.965 ± 0.018	1.032 ± 0.326	100	10
	$5 \times 6 \rightarrow 7$	1000	0.952 ± 0.020	1.086 ± 0.297	100	20
	$5 \times 6 \rightarrow 7$	1000	0.885 ± 0.029	1.506 ± 0.249	100	50
	$6 \times 7 \rightarrow 7$	1000	0.979 ± 0.019	0.974 ± 0.339	100	0
	$6 \times 7 \rightarrow 7$	1000	0.964 ± 0.018	1.034 ± 0.302	100	10
	$6 \times 7 \rightarrow 7$	1000	0.950 ± 0.021	1.082 ± 0.289	100	20
	$6 \times 7 \rightarrow 7$	1000	0.892 ± 0.028	1.380 ± 0.232	100	50
CA	$4 \times 4 \rightarrow 3$	1000	0.996 ± 0.013	0.521 ± 0.249	100	0
	$4 \times 4 \rightarrow 3$	1000	0.991 ± 0.013	0.552 ± 0.260	100	10
	$4 \times 4 \rightarrow 3$	1000	0.983 ± 0.016	0.572 ± 0.276	100	20
	$4 \times 4 \rightarrow 3$	1000	0.927 ± 0.037	0.690 ± 0.324	100	50
	$3 \times 4 \rightarrow 4$	1000	0.996 ± 0.013	0.393 ± 0.172	100	0
	$3 \times 4 \rightarrow 4$	1000	0.989 ± 0.014	0.420 ± 0.181	100	10
	$3 \times 4 \rightarrow 4$	1000	0.982 ± 0.017	0.434 ± 0.187	100	20
	$3 \times 4 \rightarrow 4$	1000	0.925 ± 0.036	0.514 ± 0.219	100	50
	$5 \times 6 \rightarrow 7$	1000	0.983 ± 0.020	0.970 ± 0.356	100	0
	$5 \times 6 \rightarrow 7$	1000	0.978 ± 0.018	1.005 ± 0.375	100	10
	$5 \times 6 \rightarrow 7$	1000	0.973 ± 0.018	1.031 ± 0.379	100	20
	$5 \times 6 \rightarrow 7$	1000	0.941 ± 0.026	1.228 ± 0.437	100	50
	$6 \times 7 \rightarrow 7$	1000	0.979 ± 0.019	0.974 ± 0.339	100	0
	$6 \times 7 \rightarrow 7$	1000	0.975 ± 0.018	1.018 ± 0.338	100	10
	$6 \times 7 \rightarrow 7$	1000	0.969 ± 0.021	1.044 ± 0.351	100	20
	$6 \times 7 \rightarrow 7$	1000	0.942 ± 0.026	1.193 ± 0.391	100	50
UD	$4 \times 4 \rightarrow 3$	1000	0.966 ± 0.068	0.904 ± 0.409	100	0
	$4 \times 4 \rightarrow 3$	1000	0.953 ± 0.063	0.931 ± 0.403	100	10
	$4 \times 4 \rightarrow 3$	1000	0.942 ± 0.058	0.939 ± 0.387	100	20
	$4 \times 4 \rightarrow 3$	1000	0.908 ± 0.047	1.035 ± 0.316	100	50
	$3 \times 4 \rightarrow 4$	1000	0.975 ± 0.063	0.676 ± 0.295	100	0
	$3 \times 4 \rightarrow 4$	1000	0.959 ± 0.058	0.690 ± 0.288	100	10
	$3 \times 4 \rightarrow 4$	1000	0.949 ± 0.058	0.711 ± 0.284	100	20

Table 5 continued

Meth.	Space dimension	Num.	Avg. AUC	Avg. RMSE	Succ. (%)	mv. (%)
	3 × 4 → 4	1000	0.902 ± 0.058	0.776 ± 0.254	100	50
	5 × 6 → 7	1000	0.934 ± 0.065	1.579 ± 0.712	100	0
	5 × 6 → 7	1000	0.933 ± 0.052	1.579 ± 0.649	100	10
	5 × 6 → 7	1000	0.925 ± 0.050	1.632 ± 0.618	100	20
	5 × 6 → 7	1000	0.899 ± 0.044	1.791 ± 0.616	100	50
	6 × 7 → 7	1000	0.926 ± 0.063	1.549 ± 0.642	100	0
	6 × 7 → 7	1000	0.924 ± 0.057	1.588 ± 0.638	100	10
	6 × 7 → 7	1000	0.915 ± 0.057	1.624 ± 0.610	100	20
	6 × 7 → 7	1000	0.896 ± 0.046	1.771 ± 0.615	100	50

For each method, space dimensions and amount of missing values (mv.), the columns show the number of utility functions (num.), average AUC and RMSE with standard deviation, and the percentage of successfully approximated functions (succ.). All the results are computed on completely defined functions

Table 6 Comparison results for DM, CA and UTADIS on real world DEX utility functions

Meth.	Avg. AUC	Avg. RMSE	Succ. (%)	Succ. unique (%)
DM	0.992 ± 0.023	1.557 ± 0.338	83.4	76.5
CA	0.996 ± 0.011	1.539 ± 0.352	82.6	75.9
UD	0.959 ± 0.110	1.717 ± 0.350	83.4	76.5

For each method, the columns show the average AUC and RMSE with standard deviation, the percentage of successfully approximated functions (succ.) and the percentage of successfully approximated unique functions (succ. unique). All the results are computed on the defined points of the functions

attribute importance in a given decision making problem. Furthermore, the weighted sum of marginal functions provides a quantitative evaluation model, which facilitates a numerical evaluation of alternatives and in this way extends the basic DEX's evaluation, which is qualitative and assigns the alternatives into discrete predefined classes.

The results presented in this paper show that the proposed methods can successfully approximate a high percentage of DEX utility functions. The accuracy varies between methods. Generally, the Conjoint analysis obtains the highest accuracy with respect to both AUC and RMSE measures. The Direct marginals method has a similar performance as the Conjoint analysis method on completely defined functions, but is more sensitive to incompleteness. This impact is particularly strong when the proportion of undefined value points crosses the 50 % threshold. At that level, the UTADIS method achieves a better accuracy than the Conjoint analysis method with respect to the AUC measure. All methods show a decline in accuracy with decreasing completeness.

The Conjoint analysis method successfully approximates a smaller percentage of functions because the current implementation returns missing values for coefficients that can not be uniquely determined. This happens when the method is applied on singular, numerically unstable, matrices. The Direct marginals method is very simple, as it constructs marginal utility functions directly from decision

rules, avoiding optimization and other extensive processing. Unfortunately, it pays the price at incomplete functions. The UTADIS method has a somewhat lower accuracy though it depends on the way in which linear programming problem is solved and the constraints are imposed. The drawback of UTADIS is that it requires the optimization direction for each attribute, which is difficult to assess particularly from non-monotone and/or highly incomplete decision rules. Anyway, we consider UTADIS to be a very good choice from the UTA family of methods for the addressed problem because it arranges alternatives into a set of ordered categories.

The example of the *Educat* function demonstrates a high accuracy of approximation of the Direct marginals and the Conjoint analysis methods on completely defined functions, however it also indicates that the user should be very careful when reasoning about the completely defined function by using only incompletely defined function with a high percentage of missing values. In this case, the Conjoint analysis and the Direct marginals method produced non-monotone marginal utility functions, which clearly breached the principle of dominance. The UTADIS method was more robust on that example, although it overestimated the utility of *Formal=PhD*.

We conclude that all the assessed methods are “fit for purpose” and can be used to approximate DEX utility functions. Among the methods, the Conjoint analysis method seems the most appropriate due to its good performance on complete DEX functions and moderate degradation of performance on incomplete ones. The Direct marginals method could be preferred for simplicity. UTADIS can be used to provide a second solution for comparison, since it outperforms the other two methods on some, especially incompletely defined, functions.

In the future work, we intend to carry out a more thorough analysis of the effects of increasing the number of categories and dimensions to the approximation accuracy. Further, we wish to investigate if the outliers in the AUC and RMSE distributions are caused by a fixed set of functions which are hard to approximate for all methods, or if each method has a distinct set that would allow improving the approximation accuracy by combining different methods. This is especially interesting for functions that contain a high percentage of missing values for which we observe distortions in the produced utility functions. The next interesting future work direction would be to guide the decision maker so as to define decision rules that would improve the definition of functions and increase the approximation accuracy. Last but not least, it would be interesting to assess the approach of Robust Ordinal Regression (Kadziński et al. 2014), which, instead of creating a single set of marginal utility functions, produces all sets compatible with the defined decision rules.

References

- Agarwal J, DeSarbo WS, Malhotra NK, Rao VR (2015) An interdisciplinary review of research in conjoint analysis: recent developments and directions for future research. *Cust Needs Solut* 2(1):19–40
- Bak A (2012) Conjoint analysis method and its implementation in conjoint R package. In: Pocięcha J, Decker R (eds) *Data analysis methods and its applications*. C.H. Beck, pp 239–248
- Błaszczczyński J, Greco S, Słowiński R, Szeląg M (2009) Monotonic variable consistency rough set approaches. *Int J Approx Reason* 50(7):979–999
- Bohanec M (2015) DEXi: Program for Multi-Attribute Decision Making, User’s Manual, Version 5.00. IJS Report DP-11897, Ljubljana: Jožef Stefan Institute

- Bohanec M, Zupan B (2004) A function-decomposition method for development of hierarchical multi-attribute decision models. *Decis Support Syst* 36:215–233
- Bohanec M, Urh B, Rajkovič V (1992) Evaluating options by combined qualitative and quantitative methods. *Acta Psychol* 80:67–89
- Bohanec M, Rajkovič V, Bratko I, Zupan B, Žnidaršič M (2013) DEX methodology: three decades of qualitative multi-attribute modelling. *Informatica* 37:49–54
- Devaud JM, Groussaud G, Jacquet-Lagrez E (1980) UTADIS: une methode de construction de fonctions d'utilite additives rendant compte de jugements globaux. European working group on MCDA, Bochum
- Ehrgott M, Figueira JR, Greco S (2010) Trends in multiple criteria decision analysis. *International series in operations research and management science*, vol 142. Springer, New York
- Fawcett T (2006) An introduction to ROC analysis. *Pattern Recogn. Lett.* 14:861–874
- Figueira JR, Greco S, Ehrgott M (2005) Multiple criteria decision analysis: state of the art surveys. Springer, Boston
- Greco S, Matarazzo B, Slowinski R (2001) Rough sets theory for multicriteria decision analysis. *Eur J Oper Res* 129(1):1–47
- Greco S, Matarazzo B, Slowiński R (2004) Axiomatic characterization of a general utility function and its particular cases in terms of conjoint measurement and rough-set decision rules. *Eur J Oper Res* 158(2):271–292
- Green PE, Srinivasan V (1990) Conjoint analysis in marketing: new developments with implications for research and practice. *J Mark* 54(4):3–19
- Hyndman RJ, Koehler AB (2006) Another look at measures of forecast accuracy. *Int J Forecast* 22(4):679–688
- Ishizaka A, Nemery P (2013) Multi-criteria decision analysis: methods and software. Wiley, Somerset
- Jacquet-Lagrez E, Siskos J (1982) Assessing a set of additive utility functions for multicriteria decision-making, the UTA method. *Eur J Oper Res* 10(2):151–164
- Kadziński M, Greco S, Slowiński R (2014) Robust ordinal regression for dominance-based rough set approach to multiple criteria sorting. *Inf Sci* 283:211–228
- Meyer P, Bigare S (2015) *Package 'MCDA'* <http://cran.r-project.org/web/packages/MCDA/MCDA> [13.05.2015]
- Mihelčić M, Bohanec M (2014) Approximating DEX utility functions with methods UTA and ACUTA. In: Piltaver R et al (eds) *Proceedings of 17th international conference information society IS 2014*, vol A. Institut Jožef Stefan, Ljubljana, pp 62–65
- Mihelčić M, Bohanec M (2015) Empirical comparison of three methods for approximating DEX utility functions. In: Zadnik Stirn L et al (eds) *Proceedings of the 13th international symposium on operational research in Slovenia*. Slovenian Society Informatika, Section for Operational Research, Ljubljana, pp 29–34
- Mileva-Boshkoska B, Bohanec M (2012) A method for ranking non-linear qualitative decision preferences using copulas. *Int J Decis Support Syst Technol* 4(2):42–58
- Ouerdane W, Maudet N, Tsoukiàs A (2010) Argumentation theory and decision aiding. In: Ehrgott M (ed) *Trends in multiple criteria decision analysis*. *International series in operations research and management science*, vol 142. Springer, New York, pp 177–208
- Robin X, Turck N, Hainard A, Tiberti N, Lisacek F, Sanchez JC, Müller M (2011) pROC: an open-source package for R and S+ to analyze and compare ROC curves. *BMC Bioinf* 12:77
- Roy B (1996) *Multicriteria methodology for decision aiding*. Kluwer Academic Publishers, Dordrecht
- Siskos Y, Grigouridis E, Matsatsinis NF (2005) UTA methods. In: Figueira JR et al (eds) *Multiple criteria decision analysis: state of the art surveys*. Springer, Boston, pp 297–343
- Yang J-B (2001) Rule and utility based evidential reasoning approach for multiattribute decision analysis under uncertainties. *Eur J Oper Res* 131(1):31–61