ECAI 2014
T. Schaub et al. (Eds.)

doi:10.3233/978-1-61499-419-0-669

Pattern-based Explanation for Automated Decisions

Ingrid Nunes[1] and **Simon Miles**[2] and **Michael Luck**[2] and **Simone Barbosa**[3] and **Carlos Lucena**[3]

Abstract. Explanations play an essential role in decision support and recommender systems as they are directly associated with the acceptance of those systems and the choices they make. Although approaches have been proposed to explain automated decisions based on multi-attribute decision models, there is a lack of evidence that they produce the explanations users need. In response, in this paper we propose an explanation generation technique, which follows user-derived explanation patterns. It receives as input a multi-attribute decision model, which is used together with user-centric principles to make a decision to which an explanation is generated. The technique includes algorithms that select relevant attributes and produce an explanation that justifies an automated choice. An evaluation with a user study demonstrates the effectiveness of our approach.

1 Introduction

Many approaches for supporting human decision making, preference reasoning or making recommendations for users have been proposed, with an underlying common goal: to choose options from those available. These approaches need user acceptance as well as efficacy to be employed in practice, and *explanations* are key to this [11]. Most forms of explanation have focused on *how* decisions are made, which makes users more tolerant to mistakes and improves *system* acceptance. However, justifying *why* particular options are chosen [12] assists users to make *better decisions* by helping them to evaluate the quality of the suggested options according to their own preferences, and to identify *refinements* that should be made in such preferences [2]. Some generic frameworks [4, 5] aim to explain automated decision making based on *multi-attribute decision models* [3], which use weights to specify trade-off among attributes, but there is a lack of evidence that they produce the explanations users need.

In order to provide guidance for explanation generation, Nunes et al. [7] performed a study investigating the explanations people give for choices they make, from which explanation guidelines and patterns were derived. In this paper, we connect this work and multi-attribute utility-based decision making approaches by proposing a technique that generates explanations based on the proposed patterns to justify *why* a particular option was chosen, and *why* other options were not. Our aim is to produce appropriate and convincing explanations. The input of our technique is a multi-attribute decision model — introduced in Section 2 — in the form of a utility function (obtained from soft-constraints and other preferences), which is used together with user-centric principles to make a decision to which an explanation is generated. We specify algorithms to select parameters required to complete explanations (Section 3), and provide a way to choose the appropriate explanation pattern in a given instance (Section 4). We evaluate our approach with a user study in Section 5, and conclude in Section 6.

[1] Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, Brazil, email: ingridnunes@inf.ufrgs.br
[2] Department of Informatics, King's College London, United Kingdom, email: {simon.miles, michael.luck}@kcl.ac.uk
[3] Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), Brazil, email: {simone, lucena}@inf.puc-rio.br

2 Multi-attribute Decision Model

Our goal is to provide an explanation for a decision, which consists of choosing an option o_c from a finite set of available options, Opt. The remaining options, $Opt_r = Opt - \{o_c\}$, are rejected. Each $o_i \in Opt$ is described in terms of a finite set of attributes, Att, where each $a_i \in Att$ is associated with a domain D_i, which establishes the values allowed for that attribute.

Users have a utility function [3] that captures their preferences, consisting of (i) utility values $v(o_i[a]) \in [-1, 1]$ (allowing the expression of both negative and positive preferences), promoted by each attribute value $o_i[a]$, and (ii) weights $w(o_i, a) \in [0, 1]$ for each attribute a that establish a trade-off relationship between attributes, where $\sum_k w(o_i, a_k) = 1$. Attribute weights are specific to each option because they may be conditioned to attribute values of an option.

We assume that weights and utility values are obtained through the use of existing elicitation techniques. Because our approach is driven by previously proposed explanation patterns [7], we must also assume that utility values are provided in the form of traditional functions or specific values, together with hard and soft constraints [6]. These are used in the explanation generation process. The latter consist of a constraint c over attribute values of a particular attribute $att(c)$. For example, c has the form $price < \$100$ and $att(c) = price$. The constraint c is associated with a utility value between $[-1, 1]$, which means that attribute values that satisfy c promote the utility value associated with $v(c)$. Extreme values (i.e., $v(c) = -1$ and $v(c) = 1$) indicate negative and positive hard constraints, meaning that options whose attribute values $o_i[a]$ either satisfy c_n, such that $v(c_n) = -1$, or do not satisfy c_p, such that $v(c_p) = 1$, should be rejected.

Moreover, option o_c is chosen not only based on a provided utility function, but also using two psychology-derived principles of how people make decisions, namely *extremeness aversion* and *trade-off contrast* [10]. The decision function $d(o_i, o_j) \mapsto [0, 1]$ evaluates whether o_i should be chosen over o_j based on a weighted sum of three factors. The first is a *cost function*, $Cost(o_i, o_j) \mapsto [0, 1]$, which indicates the disadvantages of o_i with respect to o_j quantitatively, being a weighted sum — with weights $w(o_i, a)$ — of the costs of individual attributes $AttCost(o_i, o_j, a)$. The individual attribute cost is given by

$$AttCost(o_i, o_j, a) = \begin{cases} v(o_j[a]) - v(o_i[a]) & \text{if } v(o_j[a]) > v(o_i[a]) \\ 0 & \text{otherwise} \end{cases}$$

The second is *extremeness aversion*, $ExtAversion(o_i, o_j)$, which adds a disadvantage to the option that is more extreme. Extreme op-

tions are those that compromise one attribute (low utility value) to improve another (high utility value). The option extremeness is given by the standard deviation of the utility value of individual attributes. The third is *trade-off contrast*, $ToContrast(o_i, o_j)$, which adds a disadvantage to the option that has the worst cost-benefit relationship, which is evaluated using the average cost-benefit relationship of all options. When $d(o_i, o_j) < d(o_j, o_i)$, o_i is said to be better than o_j, and the chosen option is better than all other options. Further details of how the last two factors are calculated and the selection of the best option can be found elsewhere [8].

In summary, our explanation approach requires as input: (i) hard and soft constraints, which are associated with a utility value $[1, -1]$; (ii) a utility function $v(o_i[a]) \mapsto [-1, 1]$, derived from such constraints and other preferences; and (iii) attribute weights $w(a, o)$.

3 Explanation Parameters

There are seven explanation patterns (shown below) [7], each associated with explanation templates that are parameterised by a single (for the first five patterns) or multiple (for the remaining two patterns) attributes. In this section, we show how such attributes are selected. The **Domination** pattern does not involve any parameters, so it is not discussed in this section.

- **Critical Attribute**: *chosen option was chosen because it has the best value for critical attribute .*
- **Cut-off**: *rejected option was rejected because it does not satisfy constraints associated with attribute .*
- **Domination**: *There is no reason to choose rejected option , as chosen option is better than it in all aspects.*
- **Minimum Requirements$^-$**: *Even though rejected option satisfies all your requirements, it has a worse value for attribute than chosen option .*
- **Minimum Requirements$^+$**: *Besides satisfying all your requirements, chosen option has the best value for attribute .*
- **Decisive Criteria**: *option was [chosen | rejected] because of its set of decisive attributes .*
- **Trade-off Resolution**: *Even though rejected option provides better pros than the chosen option , it has worse cons .*

3.1 Single-attribute Selection

The first single-attribute pattern concerns identifying an attribute that plays a crucial role in the decision-making process. The justification focuses only on this critical attribute, and the remaining ones are omitted. The same attribute is used to justify the chosen and all rejected options. For example, *Alice* is a student that will attend a conference. She needs to stay at the *cheapest* conference hotel with a *private room*. From the hotels listed by the conference organisers, all have private rooms, but differ in price. The cheapest is the selected option, and thus price is thus Alice's critical attribute of the decision.

In order to identify the **Critical Attribute** (if it exists) based on our input data, we use individual attribute costs, $AttCost(o_i, o_j, a)$, which provides an indication when the chosen option is preferred for this attribute w.r.t. every other option, and there is no preference between options for every other attribute. Formally, this can be expressed as follows.

Definition 1. *Let o_c be the option chosen from a set Opt. An attribute $a_{crit} \in Att$ is the **critical attribute** of the decision if, for all other options $o_r \in Opt$ where $o_r \neq o_c$, we have $AttCost(o_r, o_c, a_{crit}) > AttCost(o_c, o_r, a_{crit})$, and for all other attributes $a \in Att$ and $a_{crit} \neq a$, $AttCost(o_c, o_r, a) = AttCost(o_r, o_c, a) = 0$.*

Options that have an undesired attribute value (*cut-off value*), typically related to a hard constraint, can have their rejection justified by this attribute value — this is the case covered by the **Cut-off** pattern. To illustrate, assume that Alice provided two other preferences: she *doesn't prefer rooms where smoking is allowed* (constraint c_s such that $v(c_s) = -0.2$) and she *doesn't want a shared bathroom* (constraint c_b such that $v(c_b) = -0.8$). In addition, according to the available options, smoking is allowed in the chosen hotel H_c.

Parameters of the **Cut-off** pattern are selected according to two cases: (i) options with unsatisfied hard constraints, which specify unacceptable attribute values, regardless of the remaining attributes; and (ii) less preferred values which, though they may be compensated for other attribute values, are used as a reason to reject the option. Constraints associated with utility values 1 and -1 indicate hard constraints; consequently the first case is detected by evaluating these constraints with rejected options, and detecting their violation (unsatisfied positive hard-constraints or satisfied negative hard-constraints). In the second case, we select options that satisfy negative (or do not satisfy positive) soft-constraints, but with a restriction. The chosen option may violate a soft-constraint with this condition, e.g. hotel H_c has an attribute that Alice does not prefer (but is compensated for other attributes). Thus, it would be inconsistent to justify the rejection of a hotel "because smoking is allowed," if this is also an argument against the chosen option. So, we only select options that satisfy negative (or do not satisfy positive) soft-constraints that are *stronger* than those soft-constraints violated by the chosen option. This reasoning is compatible with the theory that states that people seek explanations to reject (and to accept) options [9]. To formalise this, we first define a function that is evaluated to true when an option o has a less preferred value according to a constraint c — $sat(o, c)$ means that o satisfies c.

$$LPV(o, c) := (sat(o, c) \wedge v(c) < 0) \vee (\neg sat(o, c) \wedge v(c) > 0)$$

Then, the strongest constraint, $stg_{cst}(C)$, of a given set of user constraints C is used to capture the strongest positive (or negative) constraint unsatisfied (satisfied) by an option.

$$stg_{cst}(o, C) := c | c \in C \wedge LPV(o, c) \wedge \forall c'.(c' \in C \wedge LPV(o, c') \wedge c \neq c' \wedge |v(c)| \geq |v(c')|)$$

Based on this strongest constraint, we detect cut-off attributes. For example, if a rejected hotel H_r has a shared bathroom, its associated explanation indicates that it was rejected because of this reason, as H_r satisfies c_b, which is stronger than c_s that is satisfied by the chosen hotel H_c. This cut-off attribute is formally defined as follows.

Definition 2. *Let $o_c, o_r \in Opt$, where o_c is the chosen option, C is a set of user constraints, and $c \in C$. An attribute $att_{co} \in Att$ is said to be a **cut-off**, or $CutOff(o_r, o_c)$, if we have:*

$$((sat(o_r, c) \wedge v(c) = -1) \vee (\neg sat(o_r, c) \wedge v(c) = 1) \vee (LPV(o_r, c) \wedge |v(c)| > stg_{cst}(o_c, C))) \wedge att(c) = a_{co}$$

If multiple attributes satisfy this property, the most important one is selected, that with the highest $w(o_r, att(c))$.

When there is a subset of options that satisfy user requirements, and one attribute is used to choose from the remaining options, **Minimum Requirements$^+$** justifies the choice, while **Minimum Requirements$^-$** explains the rejections. These patterns are applicable when users provide a set of constraints that lead to the elimination of options due to cut-off attributes (justified using the **Cut-off** pattern), allowing the identification of a *consideration set*. In addition, the chosen option has no reason to be rejected, since it satisfies all positive constraints and does not satisfy the negative ones; that is, there is no c such that $LPV(o_c, c)$. If this is the scenario in the decision making process, and one attribute, which we refer to as a

tie-breaker attribute, is decisive in choosing one option from the consideration set, we adopt these patterns to explain chosen (**Minimum Requirements**$^+$) and rejected (**Minimum Requirements**$^-$) options — excluding those rejected due to domination or cut-off attributes. The tie-breaker attribute is defined as follows.

Definition 3. *Let a_{tieBkr} and a be attributes from Att, and $o_c \in Opt$. a_{tieBkr} is said a **tie-breaker attribute**, or $TieBreaker(o_c)$, if there exists an option $o'_r \in Opt_r$ rejected due to a cut-off value, i.e. $\exists a.(CutOff(o'_r, o_c) = a)$, and for all the remaining rejected options $o_r \in Opt_r$ that $\nexists a.(CutOff(o_r, o_c) = a)$, we have $AttCost(o_r, o_c, a_{tieBkr}) > AttCost(o_c, o_r, a_{tieBkr})$. In addition, there is no a' such that $a' \neq a_{tieBkr}$ and $AttCost(o_r, o_c, a') > AttCost(o_c, o_r, a')$, i.e. a_{tieBkr} is unique.*

Now, consider all Alice's preferences mentioned in this section, a non-smoking chosen hotel H_c that is the cheapest and has a private room and bathroom, a rejected hotel H_{r_1} that is similar but more expensive than the chosen hotel, and a rejected hotel H_{r_2} that has a shared bathroom. As before, rejecting H_{r_2} is justified by a cut-off value. Given this, we observe that Alice had requirements (which excluded H_{r_2}), and the hotels H_c and H_{r_1} satisfied them. However, there is an attribute, price, which is a tie-breaker, and therefore the choice for hotel H_c is justified as it has the best value for price from the hotels satisfying Alice's requirements.

3.2 Multi-attribute Selection

One of the most important issues in the context of multi-attribute explanations is the identification of the *decisive criteria* — an issue associated with the **Decisive Criteria** pattern — of a decision. Decisive criteria consist of a subset of attributes (used as explanation) identified as the most important for preferring one option to another.

Before introducing how we identify the *decisive criteria*, we first define the concepts used in this process. When two options are compared, the pros and cons of these options with respect to each other are identified. These are captured by the sets $Att^+(o_i, o_j)$ and $Att^-(o_i, o_j)$, which are sets of attributes associated with pros and cons of o_i, respectively. $Pros(o_i, o_j)$ and $Cons(o_i, o_j)$, in turn, capture o_i's pros and cons with respect to o_j quantitatively.

Definition 4. *Let $o_i, o_j \in Opt$. Then:*

$$Att^+(o_i, o_j) = \{a | a \in Att \wedge w(o_j, a) \times AttCost(o_j, o_i, a) > 0\}$$
$$Att^-(o_i, o_j) = \{a | a \in Att \wedge w(o_i, a) \times AttCost(o_i, o_j, a) > 0\}$$

Definition 5. *Let $o_i, o_j \in Opt$. Then:*

$$Pros(o_i, o_j) = \sum_{a^+ \in Att^+(o_i, o_j)} w(o_j, a^+) \times AttCost(o_j, o_i, a^+)$$
$$Cons(o_i, o_j) = \sum_{a^- \in Att^-(o_i, o_j)} w(o_i, a^-) \times AttCost(o_i, o_j, a^-)$$

The decisive criteria are different for rejected and chosen options, discussed separately as follows.

Decisive Criteria: Rejected Options. The decisive criteria to reject an option consist of the subset of attributes whose values are sufficient to do so. For example, consider Alice's preferences, a non-smoking chosen hotel H_c that costs p, and a rejected hotel H_r where smoking is allowed and price is $\gg p$. So, regardless of the $smoking$ attribute, hotel H_r would be rejected just because of its $price$, and thus it is the decisive criterion. However, if the price of hotel H_r were $> p$, both the facts that smoking is allowed in the hotel H_r and that it is more expensive than H_c are needed to reject it. If we do not consider the benefit of H_c w.r.t. $smoking$, and cons are still higher than pros, then what matters is only the value of $price$ to choose between H_c and H_r. This intuition, which is the *keep it simple* explanation guideline [7], is formalised below. Note that different minimal subsets of attributes can be decisive, e.g. depending on the specified preferences, the attribute $smoking$ may be sufficient to reject H_c. In this case, the set of decisive criteria is the union of all these subsets, because their attributes are all relevant for justifying the rejection.

Algorithm 1: $DecisiveCriteria^-(o_r, o_c)$

Input: o_r: a rejected option; o_c: chosen option
Output: D: subset of Att containing the decisive criteria

1 $SAtt^- \leftarrow$ Sort $(Att^-(o_r, o_c), a_i \succ a_j \leftrightarrow w(o_r, a_i) \times AttCost(o_r, o_c, a_i) < w(o_r, a_j) \times AttCost(o_r, o_c, a_j))$;
2 $ACons \leftarrow 0$;
3 $Card \leftarrow 0$;
4 **while** $ACons \leq Pros(o_r, o_c) \wedge i < |Att^-(o_r, o_c)|$ **do**
5 $\quad a \leftarrow SAtt^-[Card]$;
6 $\quad ACons \leftarrow ACons + w(o_r, a) \times AttCost(o_r, o_c, a)$;
7 $\quad Card \leftarrow Card + 1$;
8 **if** $Card < |Att^-(o_r, o_c)|$ **then**
9 $\quad D, Stop \leftarrow$ DC $(\emptyset, 0, \emptyset, 0, Card, o_r, o_c, SAtt^-)$;
10 $\quad$ **if** $|D| < |Att^-(o_r, o_c)|$ **then**
11 $\quad\quad$ **return** D;
12 **return** $\emptyset$;

Definition 6. *Let $o_c, o_r \in Opt$ be the chosen and rejected options, respectively. The **decisive criteria** D for rejecting o_r is that union of all minimal (in the sense of $\subset$) subsets $S \subset Att^-(o_r, o_c)$, such that $\sum_{a \in D} w(o_r, a) \times AttCost(o_r, o_c, a) < Pros(o_r, o_c)$.*

As we need to identify different subsets of attributes, it is important to provide an efficient means of identifying them. Instead of exploring all possible subsets (which is a combinatorial problem), we propose a branch-and-bound algorithm, composed of two parts. The first, presented in Algorithm 1, finds the minimal cardinality of one possible subset that satisfies the decisive criteria property. In order to do this, we order the attributes according to their cons (from higher to lower costs) (line 1) and build a set of attributes in a stepwise fashion, accumulating their cons (lines 3–7). When we reach a set of attributes whose accumulated cons are higher than the pros, we have minimal decisive criteria. As the selected attributes are those with highest costs, there is no smaller subset of attributes that is decisive. Now that we know the cardinality of subsets we must identify, we find the other subsets of the same cardinality (lines 8–11) using Algorithm 2. Since we use the ordered attribute set of cons, we can stop our search for subsets when the first subset of attributes of that cardinality is not decisive (proofs omitted due to space constraints).

Decisive Criteria: Chosen Option. The decisive criteria of a chosen option can be either: the attribute set for which the chosen option has better values than the majority of options, and no worse for the others; or (if the former does not exist), the decisive criteria to reject the option that has the lowest pros and cons difference ("second best" option), when compared to the chosen option. In both cases, options rejected due to domination ($Expl(o, o_c) = \Psi_{dom}$) or cut-off values ($Expl(o, o_c) = \Psi_{cutOff}$) are not considered. To identify the attribute set of the first case, we define the concept of *best attributes*.

Definition 7. *Let $o_c \in Opt$ be the chosen option. The best attributes $B \subset Att$ is the set of attributes such that for all $a \in B$ and for all rejected options $o_r \in Opt_r*$ and $Opt_r* = Opt_r - \{o | Expl(o, o_c) = \Psi_{cutOff} \vee Expl(o, o_c) = \Psi_{dom}\}$, we have*

Algorithm 2: $DC(D, Idx, CAtt, ACons, Card, o_r, o_c, SAtt^-)$

Input: D: current decisive criteria, Idx: current index, $CAtt$: current attributes, $ACons$: accumulated cons, $Card$: cardinality, o_r: a rejected option; o_c: chosen option, $SAtt^-$: sorted cons
Output: D: subset of Att containing the decisive criteria, $Stop$

1 **if** $|CAtt| = Card$ **then**
2 **if** $ACons > Pros(o_r, o_c)$ **then**
3 $D \leftarrow D \cup CAtt$;
4 **return** D, $false$;
5 **else**
6 **return** D, $true$;
7 **else**
8 **for** $i \leftarrow Idx$ **to** $|SAtt^-|$ **do**
9 $a \leftarrow SAtt^-[i]$;
10 $ACons' \leftarrow ACons + w(o_r, a) \times AttCost(o_r, o_c, a)$;
11 $D, Stop \leftarrow$ `DC(`$D, i+1, CAtt \cup \{a\}, ACons', Card, o_r, o_c, SAtt^-$`)`;
12 **if** $Stop$ **then**
13 **return** D, $true$;
14 **return** D, $false$;

Algorithm 3: $DecisiveCriteria^+(o_c)$

Input: o_c: chosen option
Output: D: subset of Att containing the decisive criteria

1 $Opt_r* \leftarrow Opt - \{o | o = o_c \vee Expl(o, o_c) = \Psi_{cutOff} \vee Expl(o, o_c) = \Psi_{dom}\}$;
2 $D \leftarrow \emptyset$;
3 **foreach** $a \in Att$ **do**
4 $in \leftarrow true$;
5 $counter \leftarrow 0$;
6 **foreach** $o_r \in Opt_r*$ **do**
7 **if** $AttCost(o_r, o_c, a) = AttCost(o_c, o_r, a) = 0$ **then**
8 $counter \leftarrow counter + 1$;
9 **else if** $AttCost(o_r, o_c, a) < AttCost(o_c, o_r, a)$ **then**
10 $in \leftarrow false$;
11 **if** $in \wedge counter < \frac{|Opt_r*|}{2}$ **then**
12 $D \leftarrow D \cup \{a\}$;
13 **if** $D = \emptyset$ **then**
14 $o_{2nd_B} \leftarrow o | o \in Opt \wedge min(Pros(o_c, o) - Cons(o_c, o))$;
15 $D \leftarrow DecisiveCriteria^-(o_{2nd_B}, o_c)$;
16 **return** D;

$AttCost(o_r, o_c, a) > AttCost(o_c, o_r, a)$*, for at least* $\frac{|Opt_r*|}{2}$ *options, and* $AttCost(o_r, o_c, a) = AttCost(o_c, o_r, a) = 0$ *for the remaining ones. Moreover,* B *is maximal in the sense of* $\subset$.

We now define the decisive criteria for the chosen option, covering the two cases above. It is important to highlight that the decisive criteria for rejecting the option with the lowest pros and cons difference may not exist, as this can be less than 0, because of the trade-off contrast and extremeness aversion factors of the decision function.

Definition 8. *Let* $o_c \in Opt$ *be the chosen option. The decisive criteria* $D \subset Att$ *is the best attributes* B *of* o_c*. If* $B = \emptyset$*, then* D *is the decisive criteria of an* o_{2nd_B}*, i.e.* $DecisiveCriteria^-(o_{2nd_B}, o_c)$*, such that* $Pros(o_c, o_{2nd_B}) - Cons(o_c, o_{2nd_B})$ *is minimal, for all* $o \in Opt_r$*. Moreover,* D *exists if and only if* $|D| \neq \emptyset$.

The decisive criteria for a chosen option can be obtained by running Algorithm 3, whose first part (lines 3–12) tries to identify the best attributes; if they do not exist, the second part (lines 14–15) tries to find the decisive criteria compared to the second best option.

Trade-off Resolution. A set of attributes that are decisive criteria may not exist and, in such cases, the **Decisive Criteria** pattern cannot be applied, so the last explanation pattern — **Trade-off Resolution** — must be adopted to justify the choice to the user. Suppose Alice now has the following preferences: minimise price, distance from the conference venue and distance from tourist attractions. In addition, the chosen hotel H_c is further away from the conference venue than the rejected hotel H_r (weighted cost = 0.30), but hotel

Algorithm 4: $DecisiveProsCons(o_i, o_j)$

Input: $o_i, o_j \in Opt$
Output: P, C: subsets of Att, which represents pros and cons of o_i

1 $SortedAtt^+ \leftarrow$ `Sort (`$Att^+(o_i, o_j), a_i \succ a_j \leftrightarrow w(o_j, a_i) \times AttCost(o_j, o_i, a_i) > w(o_j, a_j) \times AttCost(o_j, o_i, a_j)$`)`;
2 $ProsLeft \leftarrow Pros(o_i, o_j)$;
3 $P \leftarrow \emptyset$;
4 $C \leftarrow \emptyset$;
5 **while** $C = \emptyset \wedge SortedAtt^+ \neq \emptyset$ **do**
6 $a \leftarrow$ `Last (`$SortedAtt^+$`)`;
7 $SortedAtt^+ \leftarrow SortedAtt^+ - \{a\}$;
8 $ProsLeft = ProsLeft - w(o_j, a) \times AttCost(o_j, o_i, a)$;
9 $P \leftarrow P \cup \{a\}$;
10 $C \leftarrow DecisiveCriteria^-(o_i, o_j, RemainingPros)$;
 `//` $DecisiveCriteria^-(o_i, o_j)$`, but considering only the remaining pros (`$ProsLeft$`)`
11 **if** $C = \emptyset$ **then**
12 $C \leftarrow Att^-(o_i, o_j)$;
13 **return** P, C;

H_r is more expensive (weighted cost = 0.18) and further away from the tourist attractions (weighted cost = 0.20). Note that there are no decisive criteria, as both cons are needed to reject H_r. In this situation we find the minimal set of attributes that are pros of hotel H_r that should not be taken into account to find decisive criteria, which in this case is distance from the conference venue. Ignoring this pro, both the other attributes satisfy the decisive criteria property, so the explanation is as follows: "even though H_r has a better distance from the conference venue than H_c, it has worse price and distance from tourist attractions." This is one possible case when there are no decisive criteria, and we next describe all the possible cases for the chosen option, and then later for the rejected options.

To explain a chosen option that does not have a set of attributes that are the decisive criteria of the decision, we have three cases to analyse, representing the three distinct reasons why there are no decisive criteria. When a chosen option o_c does not have one or more attributes that are better than the attributes of all other options, and also the pros and cons difference of the second best option is negative — that is, $Pros(o_c, o_r) < Cons(o_c, o_r)$ — meaning that the trade-off contrast and/or extremeness aversion are responsible for choosing o_c instead of o_r, we have two alternatives, which depend on the existence of a set $D \subset Att$, such that $D = DecisiveCriteria^-(o_c, o_r)$. When D exists, the provided explanation highlights that o_r has D pros (i.e. "*even though* o_r *is better considering* att_x*,* att_y*, etc.*"), and states that o_c has a better cost-benefit relationship (according to the user-centric principles). When these decisive criteria do not exist, we have a procedure to select both decisive pros and decisive cons, shown in Algorithm 4, which identifies the maximal set of pros that should be considered for enabling the existence of decisive criteria for rejecting o_c. Therefore, $DecisiveProsCons(o_c, o_r)$, for an o_r whose pros are higher than cons when compared to the chosen option, identifies the cons that should be shown in the "*even though*" part of the explanation, and also the pros that should be mentioned, which compensate cons. Moreover, the cost-benefit relationship is also highlighted since the user-centric principles play an important role in the decision.

In case o_c has the best pros and cons balance, but none of the attributes has the best values in comparison with other acceptable options (i.e. the ones not excluded due to a cut-off value or domination), we use the second best option — the option o_r that has the minimum pros and cons difference ($Pros(o_c, o_r) - Cons(o_r, o_c)$) — to explain the decision. This scenario is explained by finding the decisive criteria for rejecting the second best option, but this case was already covered in the **Decisive Criteria** pattern. Therefore, there

is only one case left where o_c has the best pros and cons balance, but there are no decisive criteria to choose it over the second best option. The explanation given in this case is based on the same algorithm adopted previously, but used in the opposite direction — $DecisiveProsCons(o_r, o_c)$ — we identify key attributes of the second best option, which are not taken into account, so that we can identify decisive criteria, and the explanation states that, even though o_r (the second best option) has better values associated with the key attributes (o_c's disadvantages), the values of the attributes that are the decisive criteria compensate for these disadvantages. These discussed cases are summarised in Table 1. Note that if we had adopted a decision making approach that did not use user-centric principles, our approach would also be applicable but, in that scenario, just the case indicated in the last row of Table 1 might occur.

The reasoning to justify rejected options is similar to that presented above. We first analyse whether the rejected option o_r has a better pros and cons balance than the chosen option ($Pros(o_r, o_c) > Cons(o_r, o_c)$). If so, the previous approach is adopted: if there is a set of attributes that characterises the decisive criteria for choosing o_r instead of o_c, i.e. $DecisiveCriteria^{-}(o_c, o_r)$, we highlight these positive aspects of o_r and state that, nevertheless, o_r has a worse cost-benefit relationship when compared to o_c; if there are no decisive criteria, we select the decisive pros and cons $\langle P, C\rangle = DecisiveProsCons(o_c, o_r)$ and, in addition to the cost-benefit relationship of o_c, we also highlight its decisive pros. This procedure is also applied when $Pros(o_r, o_c) \leq Cons(o_c, o_r)$, but no decisive criteria justify the decision.

4 Explanation Choice & Generation

After showing how parameters are selected to be part of explanations, we now present how we choose an explanation. First, we introduce the representation of each explanation type in Table 2, indicating the information needed to generate a specific explanation according to the templates proposed earlier. Domination as an explanation of a *chosen* option is our extension to the patterns, which is applied when the chosen option dominates all the others. The **Domination** pattern was reported as a pattern to justify only rejected options [7], since one option dominating all others is very unlikely to occur in practice (as options typically have pros and cons w.r.t. each other) but, since it is possible, we take it into consideration. Different explanations of Table 2 may justify choosing an option or rejecting an option. In situations in which more than one explanation is applicable, we choose one based on the following precedence: $\Psi_{crit} \triangleright \Psi_{cutOff} \triangleright \Psi_{dom^{+/-}} \triangleright \Psi_{minReq^{+/-}} \triangleright \Psi_{decisive} \triangleright \Psi_{tradeOff}$ — Ψ_{dom^+} and Ψ_{minReq^+} applies only for the chosen option; and Ψ_{dom^-}, Ψ_{minReq^-} and Ψ_{cutOff} for rejected options.

Due to space restrictions, we just provide an informal description of how to produce explanations. The main idea is to select the simplest possible explanation, for either the chosen or the rejected options. If a critical attribute guides the decision, the explanation reports this. Otherwise, the following steps are performed for the chosen option. (1) If it dominates all others, the explanation is $\Psi_{dom^+}(o_c)$. An option o_d is dominated when there is a dominant option o such that exists an attribute a where $AttCost(o_d, o, a) > 0$ and for all a' such that $a \neq a'$, $AttCost(o, o_d, a) = 0$, i.e. o_d has at least one disadvantage and no advantage with respect to o. (2) If there is a tie-breaker attribute, and there is at least one option rejected due to a cut-off value, then the explanation is based on minimum requirements. (3) If none of these cases arises, and there are decisive criteria for the choice, then the explanation is based on decisive criteria, otherwise (4) the most complex explanation is given, **Trade-off Resolution**. The process for choosing an explanation for rejected options is similar, but includes a step before the dominance test, which verifies if the option does not satisfy a cut-off value.

Table 2. Explanation Types.

Explanation Type	Parameters
Critical Attribute $\Psi_{crit}(o_c, att)$	$o_c \in Opt$ $att \in Att \wedge att = CriticalAtt(o_c)$
Domination $\Psi_{dom^+}(o_c)$ or $\Psi_{dom^-}(o_r, o_c)$	$o_c, o_r \in Opt$
Cut-off $\Psi_{cutOff}(o_r, att)$	$o_r \in Opt$ $att \in Att \wedge att = CutOff(o_r, o_c)$
Minimum Requirements$^+$ $\Psi_{minReq^+}(o_c, att)$	$o_c \in Opt$ $att \in Att \wedge att = TieBreaker(o_c)$
Minimum Requirements$^-$ $\Psi_{minReq^-}(o_r, o_c, att)$	$o_c, o_r \in Opt$ $att \in Att \wedge att = TieBreaker(o_c)$
Decisive Criteria $\Psi_{decisive}(o, target, atts)$	$o \in Opt$ $target \in \{chosen, rejected\}$ $atts \subset Att$
Trade-off Resolution $\Psi_{tradeOff}(o, target,$ $atts_P, att_C, cb)$	$o \in Opt$ $target \in \{chosen, rejected\}$ $atts_P, atts_C \subset Att$ — Pros and Cons $cb \in \{true, false\}$ Cost-benefit relationship is an argument?

5 Related Work & Evaluation

We performed a user study to evaluate our technique with users, comparing it with the existing approaches [4, 5], which have the same goal as ours and use similar input. Klein and Shortliffe [4] proposed a framework that produces explanations by identifying NOTABLY-COMPELLING attributes, those whose weighted value is above a threshold, indicating relevant pros and cons of options. Labreuche [5] proposed an approach based on the analysis of the weights, together with the utility values of the options compared, in which explanation is based on the circumstances in which a change in the weight vector changes the choice. Those attributes that impact the result of the decision are seen as the decisive criteria, used in the explanation. This is substantial progress, but there is no concrete evidence of the effectiveness of these approaches. The former approach is empirically motivated, but no study has been performed to evaluate it with users. The latter addressed a limitation of the previous work (the lack of a formal justification of why attributes should be part of explanations), but only provides an empirical evaluation of performance.

In our user study, participants provided their preferences over laptops based on an existing decision making technique [8] and its preference language. With these preferences, the decision algorithm made a choice using the decision function detailed in Section 2, resulting in a selected option from a 320 laptop catalogue, described with 58 attributes, and with the remaining ones ranked according to the decision function. Users then evaluated and compared the provided sets of explanations, in a side-by-side comparison (rotating the order of appearance), w.r.t. **transparency**: *I understand why the products were returned through the explanations in the application*; **choice quality**: *Based on the given explanations, this application made really good choices*; **trust**: *I feel that these explanations are trustworthy*; and **decision confidence**: *Based on the given explanations, I am confident that the choice made is really the best choice for me* (measurements based on an existing evaluation framework [1]). This set of questions was answered for each approach, and each question received a score according to a 7-point Likert scale. This leads to four null hypotheses: the mean of each measurement across the different approaches is the same. The study involved *30 participants* of different ages and gender, but of the same working area: we wanted

Table 1. Trade-off Resolution: selection of pros and cons to be shown in explanations for the chosen option.

Test 1	Test 2	Pros	Cons
$\exists o_r.(Pros(o_c, o_r) < Cons(o_c, o_r))$	$DecisiveCriteria^{-}(o_c, o_r) \neq \emptyset$	Cost-benefit relationship	$DecisiveCriteria^{-}(o_c, o_r)$
$\exists o_r.(Pros(o_c, o_r) < Cons(o_c, o_r))$	$DecisiveCriteria^{-}(o_c, o_r) = \emptyset$	P of $DecisiveProsCons(o_c, o_r)$ Cost-benefit relationship	C of $DecisiveProsCons(o_c, o_r)$
$\nexists o_r.(Pros(o_c, o_r) < Cons(o_c, o_r))$	$DecisiveCriteria^{-}(o_{2nd_B}, o_c) \neq \emptyset$	Decisive Criteria pattern	
$\nexists o_r.(Pros(o_c, o_r) < Cons(o_c, o_r))$	$DecisiveCriteria^{-}(o_{2nd_B}, o_c) = \emptyset$	C of $DecisiveProsCons(o_{2nd_B}, o_c)$	P of $DecisiveProsCons(o_{2nd_B}, o_c)$

participants with sufficient knowledge to judge the quality of the explanations, and thus they are computer scientists.

The user study results are shown in Figure 1 and Table 3, which shows that our approach has the best average for all measurements. Friedman's test indicated that there is a *significant* difference among the different approaches for all measurements (p-value < 0.05), so we further conducted the post-hoc tests of Wilcoxon-Nemenyi-McDonald-Thompson, which shows that the differences are due to: Klein's and our approach, for *choice quality*, and Labreuche's and our approach, for *transparency*, *trust*, and *decision confidence*.

We limit ourselves to a brief discussion of the results, given the space restrictions. We observed that the decisive criteria identification is the most important issue in explanation approaches. The attributes selected by our approach are in general preferred to those selected by Klein's approach. Although the difference between Klein's and our approach w.r.t. transparency is statistically insignificant, the variance of Klein's approach is higher, due to cases in which this approach selects *too many attributes* because of the adoption of a *fixed* threshold. Moreover, we observed that participants *liked to receive the argument related to the cost-benefit relationship*. The *complexity* of Labreuche's approach made it the least preferred among the participants, which is interesting as Labreuche's approach was proposed to address a limitation of Klein's approach. Nevertheless, in few cases in which participants were not sure about which of two options was best, they preferred Labreuche's explanation, as it provides more details about the decision. This indicates that *different levels of explanation may be provided, according to the users' needs*.

We finally highlight a performance issue. Participants lose engagement if they have to wait too long during the experiment, so a 2-minute time out (tested in a pilot study) was established for each approach to generate explanations. While our approach always executed in a short time, Labreuche's approach produced no explanation in the given time for 3 participants (discarded from the study), indicating a limitation of his approach (both approaches include branch-and-bound algorithms). Our approach had its performance also tested with data of a previous study, consisting of 113 sets of real user preferences, 144 available options and 61 attributes. Our technique took on average $125ms$ ($SD = 66.81$) on an Intel Core i5 $2.3GHz, 8GB$ of RAM to generate explanations for all options for each of user.

6 Final Remarks

In this paper, we presented a means of generating explanations for users to justify choices made based on multi-attribute decision models. Our approach consists of algorithms to identify parameters of explanation templates (part of previously proposed patterns), and to choose one of 7 possible explanation patterns to be used in a particular case. A conducted user study, involving 30 participants, indicated that our approach performs best in comparison with two existing approaches. Our future work is to extend our explanations for single-user decisions to address multi-user decision making.

Acknowledgements. Work supported by FAPERGS and CAPES.

Table 3. Explanation Results.

Measurement	Klein (K)		Labreuche (L)		Our Approach (O)	
	M	SD	M	SD	M	SD
Transparency	5.62	1.45	5.28	1.41	6.34	1.04
Choice Quality	5.17	1.46	5.17	1.36	5.76	1.40
Trust in Choice	5.48	1.30	5.34	1.17	6.17	0.93
Decision Confidence	5.10	1.40	4.76	1.48	5.45	1.48

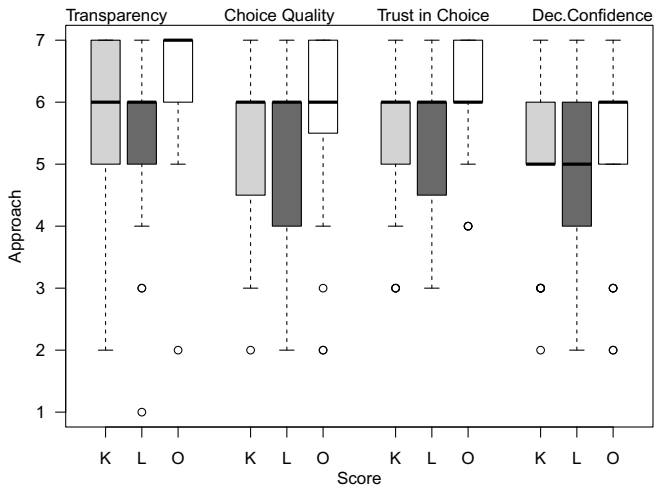

Figure 1. Measurement Scores by Explanation Approach.

REFERENCES

[1] L. Chen and P. Pu, 'User evaluation framework of recommender systems', in *SRS'10 @ IUI'10*, China, (2010). ACM.
[2] Li Chen, 'Adaptive tradeoff explanations in conversational recommenders', in *RecSys '09*, pp. 225–228, (2009).
[3] R. Keeney and H. Raiffa, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, John Wiley & Sons, Inc, 1976.
[4] D. Klein and E. Shortliffe, 'A framework for explaining decision-theoretic advice', *Artif. Intell.*, **67**, 201–243, (June 1994).
[5] C. Labreuche, 'A general framework for explaining the results of a multi-attribute preference model', *Artif. Intell.*, **175**, 1410–1448, (2011).
[6] P. Meseguer, F. Rossi, and T. Schiex, 'Soft constraints', in *Handbook of Constraint Programming*, 281–328, Elsevier, (2006).
[7] I. Nunes, S. Miles, M. Luck, and C. Lucena, 'Investigating explanations to justify choice', in *UMAP 2012*, volume 7379 of *LNCS*, pp. 212–224, Canada, (July 2012). Springer.
[8] I. Nunes, S. Miles, M. Luck, and C. Lucena, 'User-centric principles in automated decision making', in *SBIA 2012*, volume 7589 of *LNCS*, pp. 42–51, Brazil, (October 2012). Springer.
[9] E. Shafir, I. Simonson, and A. Tversky, 'Reason-based choice', in *Preference, Belief and Similarity*, 937–962, MIT, (1998).
[10] I. Simonson and A. Tversky, 'Choice in context: Tradeoff contrast and extremeness aversion', *J. of Marketing Res.*, **29**(3), 281–295, (1992).
[11] Nava Tintarev and Judith Masthoff, 'A survey of explanations in recommender systems', in *ICDEW'07*, pp. 801–810. IEEE, (2007).
[12] M. Zanker and D. Ninaus, 'Knowledgeable explanations for recommender systems', in *WI-IAT'10*, volume 1, pp. 657–660, (2010).