

Preference Modeling and Preference Elicitation: an Overview

Paolo Viappiani

(1) Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6

(2) CNRS, UMR 7606, LIP6, F-75005, Paris, France

paolo.viappiani@lip6.fr

Handling preferences [16] is important in a number of domains, including recommender systems, computational advertisement, personal cognitive assistants, systems for decision support (for example, in medicine) and robotics.

Artificial intelligence has been dealing with preferences for quite some time. In order to get closer to the goal of realizing autonomous agents that can decide and act on behalf of humans, formal tools are needed in order to model preferences, represent preferences in a compact way, support reasoning, and elicit (or learn) them from the user (decision maker). Research on preference handling systems makes use of quite a variety of different tools, including formal logic, optimization techniques from operations research; there is also a substantial intersection with research in mathematical economics, especially in approaches based on utility. An additional challenge is brought by the realization that (differently from what assumed in classical economics) humans are not rational decision makers, and are prone to decision biases; this recognition is important especially if the aim is to produce systems that are meant to be used by real users. A thorough introduction to the topic of preferences in artificial intelligence, going much more in depth, can be found in [17].

1 Modeling and Representing Preferences

Preferences express comparative judgments about elements of a set X of alternatives, choices or options. The *preference relation* $\succeq \subseteq X \times X$ is a *reflexive* binary relation; $x \succeq y$ stands for x is at least as good as y . The preference relation \succeq can be decomposed into three relations:

- Strict preference \succ : $x \succ y \wedge \neg(y \succeq x)$
- Indifference \approx : $x \succeq y \wedge y \succeq x$
- Incomparability \sim : $\neg(x \succeq y) \wedge \neg(y \succeq x)$

The triple (\succ, \approx, \sim) is the preference structure induced by \succeq . The strict preference relation is often called the asymmetric part, while indifference and incomparability comprises the symmetric part. The incomparability relation might be interpreted in different ways; two alternatives might be incomparable due to lack of knowledge (epistemic indifference) or for intrinsic reasons.

Based on different assumptions on the underlying relations, different preference structures are possible. A *total preorder* corresponds to a reflexive, complete

and transitive preference relation \succeq (the associated \succ and \approx are transitive, while \sim is empty); when \succeq is antisymmetric, \approx is the set of pairs (x, x) and we have a *total order*. A *partial preorder* is a preference structure where \succeq is reflexive and transitive (the associated \succ and \approx are transitive, while \sim is not empty); when \succeq is antisymmetric, \approx is the set of pairs (x, x) and we have a *partial order*. While often transitivity is assumed as a reasonable property, we stress that this must not necessarily be the case. In particular, the indifference relation is intransitive in semi orders [18].

If the size of X (the domain of the preference relation) is reasonable then an explicit representation of \succeq is feasible both from a cognitive (human agent) and a computational (artificial agent) point of view. However, in most cases an explicit representation is not possible and people will not provide their preference relation \succeq explicitly. A compact representation is useful so that preferences can be formulated with statements that encompass several alternatives. Preferences are most often conveyed through “preference statements”, whose interpretation (according to a precise semantics) induces a particular preference model (but the latter is not expressed directly). Languages for preference representation are evaluated with respect to a number of criteria: expressiveness, concision, cognitive relevance, computational complexity.

Among the different possible representations, we cite:

Logical representations In this approach preferences are directly expressed as logical statements. According to the classic Von Wright’s semantics [23], the statement “I prefer Φ to Ψ ” (where Φ and Ψ are formulas about the state of the world) actually means preferring the state of affairs $\Phi \wedge \neg\Psi$ (any world where Φ is true and Ψ is false) to $\Psi \wedge \neg\Phi$ (any world where Ψ is true and Φ is false). A number of “logic of preferences” have been proposed extending Von Wright’s work. Further consideration shall be given to the extent of the preferential information inferred from a statement. Let’s consider a preference statement like “I prefer red cars over to blue cars”. What does this exactly mean in term of preferential ordering? Its exact meaning (with respect to the underlying preference relation \succeq) depends on the associated semantics. Under the so-called *strong semantics*, it follows that all red cars are preferred to blue cars; while under other semantics it means that there is at least one red car that is preferred to a blue car. More precisely, under the *optimistic semantics*, the two objects in considerations are the ones that are most-preferred (maximal) according to the preference relation \succeq itself: the most-preferred red car is preferred to the most-preferred blue car (similar definitions are given for the pessimistic and the opportunistic semantics). According to the *ceteris paribus* semantics, instead, a red car is preferred to a blue car everything-else-being-equal, meaning that the two cars have the same evaluation according to the variables of interest (i.e. the brand, the engine, etc. are the same).

CP-nets The notion of *conditional preferential independence* constitutes the main building block to develop graphical models to compactly represent preferences. In a CP-net [3] each decision variable is associated with a pref-

erence table associates a given combination of values for its “parents” (a set of other variables) with a linear order establishing the preference over the domain values of the variable. We can distinguish between unconditional “Ceteris Paribus” preferences along some of the attributes of X (red is preferred to blue for the variable color all other things being the same) and conditional preferences where preferences along a certain attribute (let’s say attribute *isConvertible*) are conditioned by preferences expressed on other attributes (let’s say attribute brand and cost). Technically speaking, a CP-net is represented through a directed graph among the variables (attributes), where the maximal elements of the graph are the ones where preferences are unconditioned. CP-nets have been extended in a number of ways, notably to TCP-nets [6] in order to take into account the “relative importance” of attributes.

Utility functions and GAI-networks The intuition behind additive utility is that the contribution of each attribute to the overall utility is independent from all other attributes (and subsets of attributes): $U(x) = \sum_j u(x^j)$ where x^j is the j -th attribute of x). Under less restrictive conditions [1] we can have utility functions based on a “generalized additive independence” such that: $U(x) = \sum_i u(x^{C_i})$ where C_i are subsets of the set of attributes on which X is defined; $D_{C_i} = \prod_{j \in C_i} D^j$ is the cardinal product of the domains of the variables in C_i , and $\exists u_i : D_{C_i} \mapsto \mathbb{R}$ (a local utility value associated with an assignment of C_i). The so-called GAI networks [11] allows compact representations and efficient computation even for rather complex dependencies.

The reader interested in the subject can find a detailed introduction to the problem of preference modeling and representation in [14]. We now discuss elicitation, focusing on utility-based models.

2 Preference Elicitation

Learning or eliciting preferences means to acquire preference information in either direct or indirect way, from preference statements, critiques to examples, observations of user’s clicking behaviour, etc. The study of the assessment of the preferences of a decision maker goes back to several decades; particular emphasis has been given to the elicitation of utility functions for multi-attribute and multi-criteria settings [15]. Classic approaches for utility elicitation focus on high risk decision and aim at assessing the decision maker’s utility very precisely. The decision maker is asked a number of questions in order to assess precisely the parameters of the utility function, based on a precise *protocol*. The problem with the classic approach is that typical queries asked might be difficult to respond and the precision attained is often unnecessary.

Ideally, a system for *automated elicitation and recommendation* will only consider cognitive plausible forms of interactions, focusing on the available alternatives of the current decision problem. A number of researchers in AI [4, 9, 5, 20] have proposed the idea of an interactive utility-based recommender systems. It is assumed that the user has a latent utility function that dictates his

preferences; the system maintains a “belief” (whose nature will be clearer in a moment) about such utility function u . The general schema is as follows:

1. Some initial user preferences \mathcal{P}_0 are given; initialize belief
2. **Repeat** until the *belief* meets some termination condition
 - (a) Ask user a query q
 - (b) Observe the user response r
 - (c) Update the *belief* given r
3. Recommend the item optimal according to the current *belief*

A number of alternative proposals have been made with respect to 1) how preference uncertainty is represented in a belief, 2) which criterion is used to make a recommendation, and 3) how to select the question that is asked next. These are summarized in this table (we focus here on utility-based models).

	minimax-regret approach	maximin-utility approach	Bayesian approach
knowledge representation	constraints	constraints	prob. distribution
which option to recommend?	minimax regret	maximin utility	expected utility
which query to ask next?	worst-case regret reduction	worst-case improvement	maximin expected value of information

A possibility for representing the current belief about the utility is to encode user responses with constraints and reason about all possible consistent utility functions making use of a robust decision criterion to select the item to recommend. While maximin is a possibility [22], Boutilier et al. [5] suggest to adopt minimax regret, that is a less conservative robust criterion for decision making under uncertainty. The intuition behind the approach of *minimax regret* is that of an adversarial game; the recommender selects the item reducing the “regret” with respect to the “best” item when the unknown parameters are chosen by the adversary. The max-regret of option (or choice) x is the the maximum loss (in term of utility) that can be incurred by not choosing the (unknown) true optimal choice. The advantages of regret-based approach are threefold: 1) utility knowledge is easy to update: whenever a query is answered, we treat the answer as a new preference and derive a new feasible set of utility functions, 2) simple “priors” can be encoded with constraints in the space of utility parameters, and 3) there are efficient heuristics that directly use the computation of minimax regret to choose the queries to ask next to the user, as the *current solution strategy* [5]. The limitations of the approach are that it cannot deal with noisy responses and the formulation of the optimization is problem-dependent. See also [8] for regret-based elicitation of generalized additive utility models.

Alternatively, one could assume a Bayesian point of view: this has the advantage of handling noisy information, can exploit prior information (if available) and can be used with different assumption about the *choice model* of

the user. The belief is represented by a probability distribution over the parameters of the utility function the recommended item is the one that maximizes expected utility. When a new preference is acquired (for instance, the user states that he prefers VOLVO cars to FIAT) the distribution is updated according to Bayes (using Monte Carlo methods, or inference scheme based on expectation-propagation[13]). The problem of deciding which questions to ask could be formulated as a *POMDP* [4], however it is impractical to solve for non trivial cases. A more tractable approach is to consider (myopic) *Expected Value Of Information (EVOI)*, the difference between the expected posterior utility (of the best recommendation in the updated belief) and the current maximum expected utility. For choice queries (“Among the following options, which one do you prefer?”), Viappiani and Boutilier [21] showed that EVOI optimization is tightly connected to problem of finding an *optimal recommendation set* and near-optimal queries can be computed efficiently with worst-case guarantees. See also [7] for Bayesian elicitation of GAI models using mixture of Gaussians.

The idea of active elicitation can be applied to collaborative filtering applications, with the goal of learning the most informative ratings [10].

3 Complex Utility Models

Linear aggregators fail to represent situations where it is desired to achieve a certain degree of fairness or where one aims to model interactions between attributes. Rank-dependant aggregators sort the performance of the individual criterion by increasing order before mapping the performance vector to a scalar value (the overall utility value). The parameters provide control on: 1) the type of compromise, and 2) the attitude towards equity (fairness). Perhaps the most well known (and the simplest) of such operators is the *Ordered Weighted Average (OWA)* [24], defined as the scalar product between a vector of weights and the instantiation sorted from lowest to highest. OWA is limited as it has no notion of attribute importance; several more complex operators have then been proposed, including WOWA. The most flexible rank-dependant operator is the Choquet integral [12] that includes OWA, WOWA and the weighted sum as special case.

Eliciting or learning an utility model based on Choquet is however challenging because of the number of parameters to assess (exponential in the number of attributes). The problem can be relaxed by making additional assumptions; [19] restricts the attention to Choquet models determined by 2-additive capacities. Another approach is to focus on types of queries that are easy to handle: see [2] for the regret-based elicitation of Choquet-based utility models.

References

1. F. Bacchus and A. Grove. Graphical models for preference and utility. In *Proceedings of UAI'95*, pages 3–10, 1995.
2. Nawal Benabbou, Patrice Perny, and Paolo Viappiani. Incremental Elicitation of Choquet Capacities for Multicriteria Decision Making. In *Proceedings of ECAI'14*, pages 87–92, 2014.

3. C. Boutilier, R.I. Brafman, H.H. Hoos, and D. Poole. Reasoning with conditional ceteris paribus preference statements. In *Proceedings of UAI'99*, pages 71–80, 1999.
4. Craig Boutilier. A POMDP formulation of preference elicitation problems. In *Proceedings of AAAI'02*, pages 239–246, 2002.
5. Craig Boutilier, Relu Patrascu, Pascal Poupart, and Dale Schuurmans. Constraint-based Optimization and Utility Elicitation using the Minimax Decision Criterion. *Artificial Intelligence*, 170:686–713, 2006.
6. R. Brafman and Y. Dimopoulos. Extended semantics and optimization algorithms for CP-networks. *Computational Intelligence*, 20:219 – 245, 2004.
7. Darius Braziunas and Craig Boutilier. Local utility elicitation in GAI models. In *Proceedings of UAI'05*, pages 42–49, 2005.
8. Darius Braziunas and Craig Boutilier. Minimax regret based elicitation of generalized additive utilities. In *Proceedings of UAI'07*, pages 25–32, 2007.
9. Urszula Chajewska, Daphne Koller, and Ronald Parr. Making rational decisions using adaptive utility elicitation. In *Proceedings of AAAI'00*, pages 363–369, 2000.
10. Mehdi Elahi, Francesco Ricci, and Neil Rubens. Active learning strategies for rating elicitation in collaborative filtering: A system-wide perspective. *ACM TIST*, 5(1):13, 2013.
11. Ch. Gonzales and P. Perny. GAI networks for utility elicitation. In *Proceedings of KR'04*, pages 224 – 234, 2004.
12. Michel Grabisch and Christophe Labreuche. A decade of application of the Choquet and Sugeno integrals in multi-criteria decision aid. *Annals of Operations Research*, 175(1):247–286, 2010.
13. Shengbo Guo and Scott Sanner. Real-time multiattribute bayesian preference elicitation with pairwise comparison queries. In *AISTATS*, pages 289–296, 2010.
14. Souhila Kaci. *Working with Preferences: Less Is More*. Cognitive Technologies. Springer, 2011.
15. Ralph L. Keeney and Howard Raiffa. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. John Wiley and Sons, New York, 1976.
16. Bart Peintner, Paolo Viappiani, and Neil Yorke-Smith. Preferences in interactive systems: Technical challenges and case studies. *AI Magazine*, 29(4):13–24, 2008.
17. G. Pigozzi, A Tsoukiàs, and P. Viappiani. Preferences in artificial intelligence. Cahier du Lamsade 354, Université Paris Dauphine, 2014.
18. M. Pirlot and Ph. Vincke. *Semi Orders*. Kluwer Academic, Dordrecht, 1997.
19. Ali Fallah Tehrani, Weiwei Cheng, Krzysztof Dembczynski, and Eyke Hüllermeier. Learning monotone nonlinear models using the choquet integral. In *In Proceedings of ECML/PKDD'11*, pages 414–429, 2011.
20. Paolo Viappiani and Craig Boutilier. Regret-based optimal recommendation sets in conversational recommender systems. In *Proceedings of ACM RecSys'09*, pages 101–108, 2009.
21. Paolo Viappiani and Craig Boutilier. Optimal bayesian recommendation sets and myopically optimal choice query sets. In *Proceedings of NIPS'10*, pages 2352–2360, 2010.
22. Paolo Viappiani and Christian Kroer. Robust optimization of recommendation sets with the maximin utility criterion. In *Proceedings of ADT'13*, pages 411–424, 2013.
23. G.H von Wright. The logic of preference reconsidered. *Theory and Decision*, 3:140–169, 1972.
24. Ronald R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Trans. Systems, Man and Cybern.*, 18:183–190, 1998.