User Model User-Adap Inter (2007) 17:41–69
DOI 10.1007/s11257-006-9022-5

ORIGINAL PAPER

Efficient and non-parametric reasoning over user preferences

Carmel Domshlak · Thorsten Joachims

Received: 31 October 2005 / Accepted in revised form: 4 September 2006 /
Published online: 26 January 2007
© Springer Science+Business Media B.V. 2007

Abstract We consider the problem of modeling and reasoning about statements of ordinal preferences expressed by a user, such as monadic statement like "X is good," dyadic statements like "X is better than Y," etc. Such qualitative statements may be explicitly expressed by the user, or may be inferred from observable user behavior. This paper presents a novel technique for efficient reasoning about sets of such preference statements in a semantically rigorous manner. Specifically, we propose a novel approach for generating an ordinal utility function from a set of qualitative preference statements, drawing upon techniques from knowledge representation and machine learning. We provide theoretical evidence that the new method provides an efficient and expressive tool for reasoning about ordinal user preferences. Empirical results further confirm that the new method is effective on real-world data, making it promising for a wide spectrum of applications that require modeling and reasoning about user preferences.

Keywords Preference elicitation · Ordinal utility function · Reasoning over preferences · Support vector machines · Kernels

1 Introduction

The product catalogs of online merchants and information providers grow continuously, and with them grows the number of lay users accessing these catalogs. While keyword search provides them with some means of access to the catalogs, user needs

C. Domshlak (✉)
Faculty of Industrial Engineering and Management, Technion - Israel
Institute of Technology, Haifa 32000, Israel
e-mail: dcarmel@ie.technion.ac.il

T. Joachims
Department of Computer Science, Cornell University, Ithaca, NY 14853, USA
e-mail: tj@cs.cornell.edu

in the shopping context are typically more complex than in Web search. In particular, users have personal preferences regarding price, quality, and other attributes of the products. The aim of our work is to facilitate reasoning about such preferences to provide more targeted search results. Using the purchase of a used car as an example, we envision a catalog search-aid system that allows a user to state (using this or another form of user interface) preference statements like

1. "I like ecologically friendly cars,"
2. "For a sport car, I prefer red color over black color,"
3. "This car is better than that car,"
4. "This car would be better in red."

Alternatively, such preferences could be derived implicitly from observable user behavior (e.g., see Joachims et al. 2005). The system should use this qualitative preference information to create an effective user model, and utilize the created model to guide the user to the most relevant parts of the catalog.

In this paper, we present a new approach to modeling and reasoning about such *ordinal preferences* exemplified in the example above. We develop a robust solution for this type of modeling problem that is computationally efficient, semantically sound, and empirically effective. Specifically, we propose a method for generating an ordinal utility function from a set of qualitative preference statements. Our proposal is based on a somewhat surprising mixture of techniques from knowledge representation and machine learning. We formally show that it leads to a flexible and unprecedentedly powerful tool for reasoning about ordinal preference statements. Furthermore, we present experiments that provide initial evidence for practical applicability and effectiveness of our method, making it promising for a wide spectrum of user-centric applications.

Our approach goes beyond traditional methods for reasoning over preferences that were explored in computer science over the last three decades. Various logics of preference, graphical preference representation models, and preference learning and reasoning algorithms were proposed, in particular, in the field of artificial intelligence (AI) (for overviews, see Boutilier et al. 2004; Doyle 2004; Doyle and Thomason 1999). While these works have made significant contributions, there is still a substantial gap between theory and practice of decision support. The main problem is that so far there is no single framework for eliciting, modeling, and reasoning about user preferences that is *both* efficient and effective for *any* set of decision alternatives and *any* form of preference information. It is clear nowadays that getting closer to such a universal framework requires obtaining new insights into the problem (Doyle 2004; Shoham 1997b). Focusing on ordinal preferences, in this paper we tackle this challenge and for the first time provide a method that is both computationally efficient and without need for strong parametric assumptions. The method was first proposed in Domshlak and Joachims (2005). This paper exemplifies its use in user modeling and provides extended empirical and theoretical results.

1.1 Problem statement and background

We use the used-car catalog search as our running example. For any user willing to buy a car, the space of all possible used-car configurations constitutes the space of all possible choice alternatives Ω. The ordinal preferences of such a user can be viewed as a (possibly weak, possibly partial) binary preference relation P over Ω (Hansson

2001b). On the side of the system, the content of a used-car catalog constitutes an "available" subset of Ω that is currently relevant for the user choice, and a catalog search-aid system of our interest should allow its user to express her preferences, use the provided preference information to estimate P, and present the catalog content in a way that enables the user to quickly home in on desirable alternatives.

The content of the catalog is typically described in terms of some *attribution* $\mathbf{X} = \{X_1, \ldots, X_n\}$ (e.g., attributes of the catalog database schema), and this attribution abstracts Ω to $\mathcal{X} = \times Dom(X_i)$. For example, if the number of previous owners of a car is not kept in the catalog, and the exterior color is captured by a small-range attribute, then two car configurations in Ω that differ only in the number of previous owners or in that they are colored in different shades of blue are simply indistinguishable in this particular catalog system. As a result, the attributes $\mathbf{X}$ and their domains effectively constitute the maximal alphabet for any language that can be used to provide the catalog system with an information about Ω. In particular, the users willing to search the catalog also have to express their preferences in terms of $\mathbf{X}$.

Now, what preference information can we expect the users to provide? As suggested by multi-disciplinary literature (Chomicki 2003; Doyle 2004; Hansson 2001b; Joachims et al. 2005; Pu and Faltings 2004), in general users can (relatively) reliably provide only qualitative preference statements that either

1. compare between pairs of complete alternatives (e.g., "I prefer this car to that car"), or
2. critique certain alternatives (e.g., "I prefer a car similar to this one but without the sunroof"), or
3. generalize preference over some properties of Ω (e.g., "In a minivan, I prefer automatic transmission to manual transmission.").

Formalizing that, we assume the user provides us with a qualitative *preference expression*[1]

$$S = \{s_1, \ldots, s_m\} = \{\langle \varphi_1 \circledcirc_1 \psi_1 \rangle, \ldots, \langle \varphi_m \circledcirc_m \psi_m \rangle\}, \tag{1}$$

consisting of a set of *preference statements* $s_i = \varphi_i \circledcirc_i \psi_i$, where φ_i, ψ_i are logical formulas over $\mathbf{X}$, $\circledcirc_i \in \{\succ, \succeq, \sim\}$, and $\succ$, $\succeq$, and $\sim$ have the standard semantics of strong preference, weak preference, and preferential equivalence, respectively. For ease of presentation, in what follows we assume attributes $\mathbf{X}$ are boolean (denoting $\mathrm{Dom}(X_i) = \{x_i, \overline{x_i}\}$), and φ_i, ψ_i are propositional logic formulas. Extending our framework to arbitrary finite-domain variables is straightforward, yet requires a more involved notation that we decided to avoid here.

Given such a preference expression S, the reasoning system has to address

1. the interpretation of S in terms of the information S conveys about P,
2. a suitable representation of this information, and
3. the computational machinery to reason about this information.

Several proposals for direct logical reasoning about S have been made, yet all these proposals are limited by severe trade-offs between computational efficiency and semantic expressiveness (Boutilier et al. 2004; Lang 2004; Goldsmith et al. 2005;

[1] The actual mechanics of obtaining such preference expressions is in itself an important and complex topic that involves various issues of human-computer interaction, and in particular, user interfaces (e.g., see Chai et al. 2002; Pu et al. 2003; Blythe 2002).

Wilson 2004). In attempt to escape these trade-offs as much as possible, several works in AI (e.g., see Blythe 2002; Brafman et al. 2004; Ha and Haddawy 1999; Linden et al. 1997; McGeachie and Doyle 2004) proposed to compile information carried by S into an ordinal utility function

$$U : \mathcal{X} \mapsto \mathbb{R} \tag{2}$$

consistent with (what we believe S tells us about) P, that is requiring

$$\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}.\ \ U(\mathbf{x}) \geq U(\mathbf{x}') \Rightarrow P \not\models \mathbf{x}' \succ \mathbf{x}\,. \tag{3}$$

The consistency requirement posed on U by (3) says that ordering a set of alternatives from $\mathcal{X}$ in a non-increasing order of values provided to these alternatives by U will never put alternative x before alternative x' if our interpretation of the user's expression S implies that the user strictly prefers x' to x. In what follows, we refer to the task of constructing such a utility function U from S as *ordinal utility revelation* (OUR).

Observe that specifying a utility function U as in (2) can be expensive due to the fact that $|\mathcal{X}| = O(2^n)$. Therefore, previous works on OUR searched for special conditions on user statements under which U can be represented compactly (e.g., see Bacchus and Grove 1995; Boutilier et al. 2001; Brafman et al. 2004; Ha and Haddawy 1999; Haddawy et al. 2003; La Mura and Shoham 1999; McGeachie and Doyle 2004). The general scheme followed by these works (which we refer to as independence-based methodology) is as follows.

1. One defines certain *preferential independence* conditions on $\mathbf{X}$, and provides a "representation theorem" stating that under these conditions U can be compactly specified. Some of such foundational representation theorems come from the classical works on measurement and multi-criteria decision theories (Krantz et al. 1971; Keeney and Raiffa 1976; Fishburn 1982).
2. Next, one possibly defines some additional preferential independence conditions on $\mathbf{X}$ under which a utility function U consistent with S is not only compactly representable, but also can be efficiently generated from S (Ha and Haddawy 1999; McGeachie and Doyle 2004; Brafman et al. 2004).
3. Finally, the system poses restrictions on the form of allowable preference expressions, so that they are constrained to a sufficiently simple language for which the conditions from above are fulfilled.

Considering the first step of independence-based methodology, note that any function U as in (2) can be represented in a generalized additive form (Fishburn 1982; Bacchus and Grove 1995):

$$U(\mathbf{x}) = \sum_{i=1}^{k} u_i(\mathbf{x}[i]) \tag{4}$$

where each sub-function u_i depends on some subset of variables $\mathbf{X}[i] \subseteq \mathbf{X}$, and $\mathbf{x}[i]$ is the restriction of a complete assignment $x \in \mathcal{X}$ to the attributes in $\mathbf{X}[i]$. Note that u_i can be selected to be any function that maps a (partial) assignment to a real value. The generality of (4) is immediate since we may have $k = 1$ and $\mathbf{X}[i] = \mathbf{X}$, which means that one can effectively select a function $u_1(\mathbf{x})$ that assigns an arbitrary utility to each possible complete assignment $\mathbf{x} \in \mathcal{X}$. While by itself this generality has no practical implications, (4) allows us to generalize the core assumptions of previous works on

OUR—each concrete instance of the independence-based methodology corresponds to some independence conditions on $\mathbf{X}$ that guarantee existence of U (a) satisfying (3) and (b) having a *compact* decomposition in terms of (4) (that is, defined by a small number k of small attribute subsets). An additional key property of all previous works is that attribute subsets $\mathbf{X}[1], \ldots, \mathbf{X}[k]$ decomposing U are assumed to be known to the system.[2]

Finally, to our knowledge, all the works on independence-based OUR (except for the approach suggested in Haddawy et al. 2003) assume that the user's preference expression is consistent, that is, the user makes no mistakes in specifying her preferences. In practice, however, this assumption is not necessarily reasonable, and later we discuss this issue with respect to our approach.

1.2 Challenges and our results

In short, computationally efficient schemes for multi-attribute utility revelation proposed in economics and AI are parametrized by the *structure* that user preferences induce on $\mathbf{X}$, and thus are applicable only when such *compact structure exists* and is *known* to the system. One of the problems though is that the attributes $\mathbf{X}$ used by the catalog system do not necessarily correspond to the criteria affecting preference of each individual user over the actual physical alternatives in Ω. Hence, even if user preferences are compactly structured, they should not necessarily be compactly structured *over* $\mathbf{X}$.

Having in mind these limitations, we list the challenges that search/decision support applications pose to the research on OUR. The vision here is threefold.

1. The user should be able to provide preference expressions S while being as little constrained in her language as possible.
2. The utility revelation machinery (that is, generating U from S) should be completely non-parametric, that is free of any explicit assumptions about the structure of the user's utility function.
3. Both utility revelation and using the revealed utility function should be computationally efficient, including the case where user preferences pose no significant independence structure on $\mathbf{X}$ whatsoever.

To the best of our knowledge, in this paper we present the first approach that fulfills these goals. Combining ideas from knowledge representation, machine learning, and philosophical logic we provide a concrete mathematical setting in which all the above desiderata can be successfully achieved, and formally show that this setting is appealing both semantically and computationally. The mathematical framework we propose is based on a novel high-dimensional structure for preference decomposition, and a specific adaptation of certain standard techniques for high-dimensional continuous optimization, frequently used in machine learning in the context of Support Vector Machines (SVMs) (Vapnik 1998).

[2] In fact, most systems to date (except for a few recent development in Brafman et al. 2004; Boutilier et al. 2005; Haddawy et al. 2003) take a very simplifying assumption that user preferences can always be represented to a large degree by a linear function over $\mathbf{X}$ (that is, $k = n$, and $\mathbf{X}_i = \{X_i\}$) (Blythe 2002; Burke et al. 1997; Linden et al. 1997; Pu and Faltings 2004; Shearin and Lieberman 2001).

2 High-dimensional preference decomposition

Considering our vision for modeling and reasoning about user's ordinal preferences, one can certainly be somewhat skeptical. Indeed, how can OUR be efficient if the user preferences pose no significant independence structure on $\mathbf{X}$, or, if they do, the system is not provided with this independence information? In the rest of this section we describe the representational and semantic sides of our proposal, and then discuss its computational properties in Sect. 3.

2.1 Representation model

The very basic idea underlying our proposal is as simple as it gets: *Since we are not provided with useful preferential independence information in the original representation space $\mathcal{X}$, maybe we should move to a different space in which no such independence information is required?* Proceeding with the technical side of this idea, let us schematically map the alternatives $\mathcal{X}$ into a new, higher dimensional space $\mathcal{F}$ using a certain mapping

$$\Phi : \mathcal{X} \mapsto \mathcal{F} = \mathbb{R}^{4^n}. \tag{5}$$

As one would expect, the mapping Φ is not arbitrary, and it establishes a clear connection between the dimensions of $\mathcal{X}$ and $\mathcal{F}$ as follows. Let $\mathbf{F} = \{\mathfrak{f}_1, \ldots, \mathfrak{f}_{4^n}\}$ be a labeling of the dimensions of $\mathcal{F}$, and

$$\mathscr{D} = \bigcup_{i=1}^{n} \mathrm{Dom}(X_i) = \{x_1, \overline{x_1}, \ldots, x_n, \overline{x_n}\},$$

be the union of attribute domains in $\mathbf{X}$. Let $\mathsf{val} : \mathbf{F} \to 2^{\mathscr{D}}$ be a bijective mapping from the dimensions of $\mathcal{F}$ onto the power set of $\mathscr{D}$, uniquely associating each dimension $\mathfrak{f}_i$ with a subset $\mathsf{val}(\mathfrak{f}_i) \subseteq \mathscr{D}$, and vice versa. Likewise, in what follows, by $\mathsf{Var}(\mathfrak{f}_i) \subseteq \mathbf{X}$ we denote the subset of attributes "instantiated" by $\mathsf{val}(\mathfrak{f}_i)$. For example, if $\mathsf{val}(\mathfrak{f}_i) = \{x_2, \overline{x}_3, x_{17}\}$, then $\mathsf{Var}(\mathfrak{f}_i) = \{X_2, X_3, X_{17}\}$. Given that, for each $\mathbf{x} \in \mathcal{X}$ and $\mathfrak{f}_i \in \mathbf{F}$, we set:

$$\Phi(\mathbf{x})[i] = \begin{cases} 1, & \mathsf{val}(\mathfrak{f}_i) \neq \emptyset \wedge \mathsf{val}(\mathfrak{f}_i) \subseteq \mathbf{x} \\ 0, & \text{otherwise} \end{cases}. \tag{6}$$

To illustrate this mapping Φ, let $\mathcal{X}$ be a two-dimensional space defined by an attribution $\mathbf{X} = \{X_1, X_2\}$, and let $\mathbf{x} = x_1\overline{x_2}$. From (6), for this $\mathbf{x}$ we have $\Phi(\mathbf{x})[i] = 1$ if and only if $\mathsf{val}(\mathfrak{f}_i) \in \{x_1, \overline{x_2}, x_1\overline{x_2}\}$, that is

$$\Phi(\mathbf{x}) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \begin{array}{l} \mathsf{val}(\mathfrak{f}_1) = x_1 \\ \mathsf{val}(\mathfrak{f}_2) = \overline{x_1} \\ \mathsf{val}(\mathfrak{f}_3) = x_2 \\ \mathsf{val}(\mathfrak{f}_4) = \overline{x_2} \\ \mathsf{val}(\mathfrak{f}_5) = x_1x_2 \\ \mathsf{val}(\mathfrak{f}_6) = x_1\overline{x_2} \\ \mathsf{val}(\mathfrak{f}_7) = \overline{x_1}x_2 \\ \mathsf{val}(\mathfrak{f}_8) = \overline{x_1x_2} \end{array} \tag{7}$$

and $\Phi(\mathbf{x})[i] = 0$, otherwise. Note that the (ignored in 7) dimensions $\mathfrak{f}_9, \ldots, \mathfrak{f}_{16}$ either have $\mathsf{val}(\mathfrak{f}_i) = \emptyset$, or $\mathsf{val}(\mathfrak{f}_i)$ contains both a literal x_j and its negation $\bar{x}_j$. It is not hard

to verify that these dimensions are essentially redundant because for such $\mathfrak{f}_i$ we have $\Phi(\mathbf{x})[i] = 0$ for *all* alternatives' descriptions $\mathbf{x} \in \mathcal{X}$. Indeed, later we show that we actually use only the $(3^n - 1)$-dimensional subspace of $\mathcal{F}$, dimensions of which correspond to all the *non-empty partial assignments to* $\mathbf{X}$. In fact, for ease of presentation, in what follows we discuss $\mathcal{F}$ as if ignoring its redundant dimensions. However, for some technical reasons important for our computational machinery (and clarified in Sect. 3), the structure of $\mathcal{F}$ and Φ has to be defined as in (5–6).

Geometrically, Φ maps each n-dimensional vector $\mathbf{x} \in \mathcal{X}$ to the 4^n-dimensional vector in $\mathcal{F}$ that uniquely encodes the set of all projections of $\mathbf{x}$ onto the subspaces of $\mathcal{X}$. This, however, does not shed too much light on the semantics of Φ. To see the latter, recall that

1. The attribution $\mathbf{X}$ is just one out of many possible attributions of the physical alternatives Ω (induced by some system-specific considerations), and as such it does not necessarily correspond to the criteria affecting preferences of a user over Ω.
2. The description of the alternatives from Ω in the catalog in terms of the attribution $\mathbf{X}$ forces the user to express her preferences only in terms of $\mathbf{X}$.
3. However, if the user does articulate some preference information in terms of $\mathbf{X}$, then the implicit preference-related criteria behind this information obviously have *some* encoding in terms of $\mathbf{X}$.

To stress the latter point, suppose that the catalog system maintains in its records the class and the color of the cars as a pair of attributes $X, X' \in \mathbf{X}$. Suppose that a user of the system likes sport cars, likes green color, and yet strongly dislikes green sport cars, providing a statement capturing the latter piece of information. In this case, the user articulates her preferences over a *combination of a particular pair of values of* X *and* X', and no single attribute directly maintained by the system can capture the user's utility alone. However,

1. if such a complex preference is articulatable in terms of the attributes $\mathbf{X}$, then it has to correspond to this or another set of value assignments to a subset of $\mathbf{X}$, and
2. the evaluation of *any* abstracted alternative $\mathbf{x} \in \mathcal{X}$ with respect to such a complex preference corresponds to a single (possibly empty) value assignment to a subset of $\mathbf{X}$, and that is, to a *single dimension of* $\mathcal{F}$.

This means that there are not only dimensions in $\mathcal{F}$ that capture the utilities of "sports car" and "green," but there is also a dimension that directly captures the utility of "green sports cars." This correspondence between complex preferences and individual dimensions of $\mathcal{F}$ makes the semantics of $\mathcal{F}$ very attractive. In addition, Theorem 1 below shows that $\mathcal{F}$ is not only semantically attractive, but also satisfies our desire for "no need for independence information."

Theorem 1 *Any preference ordering P over $\mathcal{X}$ is additively decomposable in $\mathcal{F}$. That is, for any preference ordering P over $\mathcal{X}$, there exists a* linear *function*

$$\mathfrak{U}(\Phi(\mathbf{x})) = \sum_{i=1}^{4^n} w_i \, \Phi(\mathbf{x})[i] \tag{8}$$

with weights $w_i \in \Re$ that satisfies (3).

Proof The proof of Theorem 1 is rather straightforward.[3] Since the binary relation P over $\mathcal{X} \times \mathcal{X}$ is an ordering, there exists a utility function U satisfying (3) with respect to P, and let $\mathbf{X}[1], \ldots, \mathbf{X}[k]$ be the factors of U with respect to its generalized additive form $U(\mathbf{x}) = \sum_{i=1}^{k} u_i(\mathbf{x}[i])$ as in (4).

Let the parameter weights $w_1, \ldots, w_{4^n} \in \mathbb{R}$ be defined as:

$$w_i = \begin{cases} u_j(\mathsf{val}(\mathfrak{f}_i)), & \mathsf{Var}(\mathfrak{f}_i) = \mathbf{X}[j] \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

and let

$$\mathfrak{U}(\Phi(\mathbf{x})) = \sum_{i=1}^{4^n} w_i \, \Phi(\mathbf{x})[i]. \tag{10}$$

By the construction, we have $\mathfrak{U}(\Phi(\mathbf{x})) = U(\mathbf{x})$, and thus:

$$\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}. \quad \mathfrak{U}(\Phi(\mathbf{x})) \geq \mathfrak{U}(\Phi(\mathbf{x}')) \Rightarrow P \not\models \mathbf{x}' \succ \mathbf{x}. \tag{11}$$

□

Since, by Theorem 1, the dimensions of $\mathcal{F}$ can successfully "linearize" any preference ordering P over $\mathcal{X}$, in what follows we can focus only on linear utility functions as in (8). Of course, the reader may rightfully wonder whether this linearization in a space of dimension 4^n can be of any practical use, and not just a syntactic sugar. At this stage, however, we ask the reader to postpone the computational concerns, and focus on the *interpretation* of preference expressions in terms of this high-dimensional space $\mathcal{F}$.

2.2 Interpretation of preference expressions

Considering qualitative preference statements forming user expressions S as in (1), there are two major categories of preference statements one would certainly like to allow in S (Hansson 2001b):

1. dyadic (comparative) statements, indicating a relation between two referents using the concepts such as "better," "worse," and "equal in value to," and
2. monadic (classificatory) statements, evaluating a single referent using ordinal language concepts such as "good," "very bad," and "worst."

One can raise expectations even further by asking to allow, for instance, "higher order" preferences such as "x is preferred to y more than z is preferred to w" (Packard 1975). While we strive to support all these forms of preference statements and more, for ease of presentation, let us focus on dyadic statements for now.

Let us first consider an "instance comparison" statement "$\mathbf{x}$ is better than $\mathbf{x}'$", where $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$. The interpretation of this statement poses no serious difficulties because it explicitly compares between complete descriptions of two alternatives. However, this is the exception, rather than the rule. Most of the preference statements that we use in our everyday activities (e.g., "I prefer compact cars to SUVs") have this or another *generalizing* nature. As such, these statements typically mention only a subset of attributes. This creates an ambiguity with respect to the actual referents of

[3] It is worth noting here that this explicit "construction" of $\mathfrak{U}$ only serves the *existential* proof of Theorem 1, and does not reflect whatsoever the machinery of our proposal presented later in the paper.

these statements. Several proposals on how to interpret generalizing preference statements have been made both in philosophy and AI. However, there is no agreed-upon solution to this problem (e.g., see Boutilier 1994; Doyle and Wellman 1994; Hansson 2001a), and the inherent uncertainty about the message that such a statement aims to communicate makes developing such a solution very questionable. On the other hand, different existing proposals are not entirely tangential. Specifically, all these proposals suggest to interpret generalizing preference statements as indirectly comparing between sets of alternatives from Ω, while possibly disagreeing on what sets of alternatives are actually compared by each statement separately, and/or by a multi-statement preference expression as a whole.[4]

Considering interpretation of qualitative preference expressions in our new representation space $\mathcal{F}$, observe that each parameter w_i of $\mathfrak{U}$ in (8) can be seen as representing the *marginal utility* of the interaction between the attributes $\mathsf{Var}(\mathfrak{f}_i) \subseteq \mathbf{X}$ when these take the value $\mathsf{val}(\mathfrak{f}_i)$. In other words, w_i represents the marginal contribution of the "criterion" $\mathsf{val}(\mathfrak{f}_i)$ to attractiveness (or unattractiveness) of any alternative $\mathbf{x}$ such that $\mathsf{val}(\mathfrak{f}_i) \subseteq \mathbf{x}$. Moreover, the dimensional structure of $\mathcal{F}$ allows us to devote w_i to this specific criterion *only*; the marginal utilities of all the syntactically related criteria [captured by subsets and supersets of $\mathsf{val}(\mathfrak{f}_i)$] can be successfully represented by other parameters w of $\mathfrak{U}$.

Having this perspective in mind, let us consider an arbitrary dyadic statement $\varphi \succ \psi$, and begin with providing some essential notation. Let $\mathbf{X}_\varphi \subseteq \mathbf{X}$ (and similarly $\mathbf{X}_\psi$) be the variables involved in φ, and $M(\varphi) \subseteq \mathrm{Dom}(\mathbf{X}_\varphi)$ be the set of all φ's models in the subspace of $\mathcal{X}$ defined by $\mathbf{X}_\varphi$. For instance, if $\mathbf{X} = \{X_1, \ldots, X_{10}\}$, and $\varphi = X_1 \vee X_2$, then $\mathbf{X}_\varphi = \{X_1, X_2\}$, and $M(\varphi) = \{x_1x_2, \overline{x_1}x_2, x_1\overline{x_2}\}$. To avoid confusion between $M(\varphi)$ and the standard notion of "models of φ" (which are *complete* assignments to $\mathbf{X}$ that satisfy φ), in what follows we refer to the elements of $M(\varphi)$ as *local models* of φ.

Following the most standard (if not the only) interpretation scheme for OUR, we compile $\varphi \succ \psi$ into a set of constraints on the space of candidate real-valued utility functions (Krantz et al. 1971). However, in contrast to previous works, in our case these constraints are posed not on a space of functions from the original attribute-based space $\mathcal{X}$, but on the space of functions of form (8), that is, the space of linear, real-valued functions from our new representation space $\mathcal{F}$. Specifically, we compile the statement $\varphi \succ \psi$ into a set of $|M(\varphi)| \times |M(\psi)|$ constraints[5]

$$\forall \mathbf{m}_\varphi \in M(\varphi), \forall \mathbf{m}_\psi \in M(\psi). \quad \sum_{\mathfrak{f}_i : \mathsf{val}(\mathfrak{f}_i) \in 2^{\mathbf{m}_\varphi}} w_i \;>\; \sum_{\mathfrak{f}_j : \mathsf{val}(\mathfrak{f}_j) \in 2^{\mathbf{m}_\psi}} w_j, \tag{12}$$

where $2^{\mathbf{m}}$ denotes the set of all non-empty value subsets of the local model $\mathbf{m}$. For example, statement $(X_1 \vee X_2) \succ (\neg X_3)$ (e.g., "It is more important that the car is powerful or fast than not having had an accident") is compiled into

$$\begin{aligned} w_{x_1} + w_{x_2} + w_{x_1x_2} &> w_{\overline{x_3}} \\ w_{x_1} + w_{\overline{x_2}} + w_{x_1\overline{x_2}} &> w_{\overline{x_3}} \\ w_{\overline{x_1}} + w_{x_2} + w_{\overline{x_1}x_2} &> w_{\overline{x_3}}. \end{aligned} \tag{13}$$

[4] For an excellent survey of this topic, we refer the reader to Hansson (2001b).

[5] The constraints for dyadic statements of the form $\varphi \succeq \psi$ and $\varphi \sim \psi$ are similar to (12) with $>$ being replaced by $\geq$ and $=$, respectively.

The constraint system $\mathfrak{C}$ resulting from such compilation of a user expression S defines the space of solutions for our formulation of OUR. On the side of the semantics, we argue that $\mathfrak{C}$ corresponds to a *least committing interpretation* of preference statements. Specifically, the three guiding principles of this interpretation are as follows.

A1 If $\langle \varphi \succ \psi \rangle \in S$, then every local model $\mathbf{m}_\varphi$ of φ is preferred to every local model $\mathbf{m}_\psi$ of ψ.
A2 All possible explanations for $\varphi \succ \psi$ should be considered.
A3 If there is no reason for a bias toward certain explanations for $\varphi \succ \psi$, a most general explanation should be preferred.

The principle (A1) with respect to a preference statement $\varphi \succ \psi$ is enforced by the universal quantification in (12). Let us now lift our mapping Φ to

$$\Phi : \mathcal{X}^+ \mapsto \mathcal{F}\,, \tag{14}$$

where $\mathcal{X}^+$ is the extension of $\mathcal{X}$ to all *partial* assignments to the attributes $\mathbf{X}$. Note that this lifting of applicability of Φ from $\mathcal{X}$ to $\mathcal{X}^+$ is completely innocuous and it does not require changes even in (6). Under the extended Φ, by adopting (A1) we lift the semantics of the utility functions $\mathfrak{U}$ in (8) to partial assignments to $\mathbf{X}$, making (12) equivalent to

$$\forall \mathbf{m}_\varphi \in M(\varphi), \forall \mathbf{m}_\psi \in M(\psi). \quad \mathfrak{U}(\Phi(\mathbf{m}_\varphi)) > \mathfrak{U}(\Phi(\mathbf{m}_\psi)). \tag{15}$$

Intuitively, (15) encodes the assumption that if the user has to choose between two alternatives $\mathbf{x}$ and $\mathbf{x}'$ while knowing *only* that $\mathbf{x}$ satisfies (= contains) $\mathbf{m} \in M(\varphi)$ and $\mathbf{x}'$ satisfies $\mathbf{m}' \in M(\psi)$, then the user will choose $\mathbf{x}$. To us this assumption appears to be the least one can assume about the actual message behind the statement $\varphi \succ \psi$.

Here we note that even a slightly *more* committing interpretation has been considered natural in the philosophical literature on preference logics. Specifically, let φ/ψ be equal to φ if $\varphi \wedge \neg\psi$ is logically contradictory (that is, $\varphi \rightarrow \psi$), and equal to $\varphi \wedge \neg\psi$, otherwise. The standard interpretation of an informal user statement "φ is preferred to ψ" is that local models of φ/ψ are preferred to local models of ψ/φ (Packard 1975; von Wright 1972; Hansson 2001a). In the scope of our approach, this translates into requiring

$$\forall \mathbf{m}_{\varphi/\psi} \in M(\varphi/\psi), \forall \mathbf{m}_{\psi/\varphi} \in M(\psi/\varphi). \quad \mathfrak{U}(\mathbf{m}_{\varphi/\psi}) > \mathfrak{U}(\mathbf{m}_{\psi/\varphi}) \tag{16}$$

instead of (15). If chosen, however, this change in the interpretation will have no impact on the rest of our approach as (16) will be treated as if the user explicitly provides us with a statement "φ/ψ is preferred to ψ/φ."

Considering the principle (A2), notice that the statement $\varphi \succ \psi$ provides us with some information on user preferences, but not with an explanation to this preference. On the other hand, different explanations for why the user states $\varphi \succ \psi$ are possible, and in fact, these explanations may be not the same for different pairs of local models $\mathbf{m} \in M(\varphi), \mathbf{m}' \in M(\psi)$. For instance, it is *possible* in our example that the user prefers $x_1 \wedge \overline{x_2}$ to $\overline{x_3}$ mainly due to the high marginal value of the criterion x_1 and $\overline{x_1} \wedge x_2$ to $\overline{x_3}$ mainly due to the high marginal value of the criterion $\overline{x_1}x_2$. This is why each constraint in (12) does not make unnecessary commitments, constraining only the relation between the *cumulated* marginal utilities of the criteria $\mathsf{val}(\mathfrak{f}_i) \in 2^{\mathbf{m}_\varphi}$ and $\mathsf{val}(\mathfrak{f}_j) \in 2^{\mathbf{m}_\psi}$ involved in the local models of φ and ψ, respectively.

Finally, while different explanations for each statement $\langle \varphi \succ \psi \rangle \in S$ separately, and the expression S as a whole, are possible, at the end the user-modeling system has to adopt and act upon one such explanation. On this matter, (A3) simply states the principles of Bayesian inference that advice to prefer the least biased explanation possible. In Sect. 3 we describe how we pick a particular utility function $\mathfrak{U}$ for a given set of constraints, and justify this selection procedure with respect to (A3) in Sect. 4. This finalizes our arguments for the "least committing" of our interpretation scheme, and we hope to have convinced the reader that semantically this scheme is appealing. What still remains to be shown, however, is that our scheme is also computationally realistic, and we consider this issue in the next section.

3 Computational machinery

Returning to the computational side of the story, it is not hard to see that the structure of the target utility functions $\mathfrak{U}$, as well as the constraint system $\mathfrak{C}$ built according to (12/15), pose numerous complexity issues.

(a) Our target utility function $\mathfrak{U}$ is a linear, real-valued function from a 4^n dimensional space $\mathcal{F}$. Thus, not only generating $\mathfrak{U}$, but even keeping and evaluating this function explicitly might be infeasible. Likewise, though the constraint system $\mathfrak{C}$ is linear, it is linear in the exponential space $\mathbb{R}^{4^n}$. Hence, in general, the dimensionality of $\mathfrak{C}$ does not allow us to solve it directly.

(b) The summations in *each* constraint as in (12) are exponential in the arity of φ and ψ (that is, in $|\mathbf{X}_\varphi|$ and $|\mathbf{X}_\psi|$). That is, not only the dimensionality of $\mathfrak{C}$, but also its description complexity is an issue. Practice shows that this source of worst-case complexity is something we should worry about because it is far from being of theoretical interest only. For instance, each "instance comparison" between a pair of complete alternatives in $\mathcal{X}$ is translated into a (single) constraint with up to 2^{n+1} summation terms, and this is a very natural form of everyday preference statements, already supported by some catalog search-aiding systems.

(c) The number of constraints generated for each preference statement can also be exponential in the arity of φ and ψ. This issue, however, seems to be significantly less problematic in practice because the number of constraints actually equals the number of local models of φ and ψ. In other words, an exponentially large number of constraints may come only from a preference statement comparing between disjunctive formulas with numerous disjuncts, that is, a preference statement that simultaneously compares between large sets of models. While possible in principle, we believe that such preferential comparisons are rarely natural.

(d) Finally, strictly speaking, even if the number of local models of φ and ψ is small, the actual enumeration of these models required by (12) can be computationally intractable due to #P-completeness of propositional model counting. However, we believe that propositional formulas coming from user statements are unlikely to be of a size that would pose computational challenges in practice.

From the theoretical perspective, all these complexity issues are equally important. However, as argued above, the first two sources of complexity appear to be the most crucial in practice. Fortunately, here we show that these two issues can be overcome.

In what follows we introduce the computational machinery underlying our framework. As in the previous section, consider a dyadic statement $\varphi \succ \psi$. Our translation of this statement leads to a set of $|M(\varphi)| \times |M(\psi)|$ linear constraints of the form:

$$\begin{aligned}\forall \mathbf{m}_\varphi \in M(\varphi),\ \forall \mathbf{m}_\psi \in M(\psi):&\\ \mathfrak{U}\left(\Phi(\mathbf{m}_\varphi)\right) > \mathfrak{U}\left(\Phi(\mathbf{m}_\psi)\right) &\Leftrightarrow \sum_{i=1}^{4^n} w_i\ \Phi(\mathbf{m}_\varphi)[i] > \sum_{i=1}^{4^n} w_i\ \Phi(m_\psi)[i]\\ &\Leftrightarrow \mathbf{w}\cdot\Phi(\mathbf{m}_\varphi) > \mathbf{w}\cdot\Phi(\mathbf{m}_\psi) \qquad (17)\end{aligned}$$

From (17), the set of utility functions consistent with an expression of κ such preference statements $\{\langle\varphi_1 \succ \psi_1\rangle, \ldots, \langle\varphi_\kappa \succ \psi_\kappa\rangle\}$ is defined by the solutions of the linear system $\mathfrak{C}$:

$$\begin{aligned}\forall 1 \le i \le \kappa,\ \forall \mathbf{m}_{\varphi_i} \in M(\varphi_i),\ \forall \mathbf{m}_{\psi_i} \in M(\psi_i):&\\ \mathbf{w}\cdot\Phi(\mathbf{m}_{\varphi_i}) > \mathbf{w}\cdot\Phi(\mathbf{m}_{\psi_i}),& \qquad (18)\end{aligned}$$

consisting of $k = \sum_{i=1}^{\kappa} |M(\varphi_i)| \times |M(\psi_i)|$ constraints in $\mathbb{R}^{4^n}$. To simplify the presentation, in what follows we denote the k ordered pairs of local models underlying (19) simply by $(\mathbf{m}_i, \mathbf{m}'_i)$, $1 \le i \le k$, rewriting (18) as

$$\forall 1 \le i \le k\colon\ \mathbf{w}\cdot\Phi(\mathbf{m}_i) > \mathbf{w}\cdot\Phi(\mathbf{m}'_i). \qquad (19)$$

Clearly, naive approaches to solving such systems (that is, finding a $\mathbf{w} \in \Re^{4^n}$ that satisfies all the constraints) will be computationally intractable for any non-trivial number of attributes $n = |\mathbf{X}|$. However, below we show that exploiting duality techniques from optimization theory (Bertsekas et al. 2003) and Reproducing Kernel Hilbert Spaces (RKHS) (Kimeldorf and Wahba 1971; Wahba 1990; Vapnik 1998) allows solving such systems in time only linear in n and polynomial in k.

At the first step, we reformulate our task of satisfying $\mathfrak{C}$ as an optimization problem. Since the solution of (19) is typically not unique, we select a particular solution by adding an objective function and a "margin" by which the inequality constraints should be fulfilled. Specifically, similar to an ordinal regression Support Vector Machine (SVM) (Herbrich et al. 2000), we search for the smallest L_2 (i.e., shortest Euclidian length) weight vector $\mathbf{w}$ that fulfills all constraints with margin 1. The corresponding constrained optimization problem is:

$$\begin{aligned}&\text{Minimize } (w.r.t.\ \mathbf{w})\colon\ \frac{1}{2}\mathbf{w}\cdot\mathbf{w}\\ &\text{subject to:}\\ &\quad \forall 1 \le i \le k:\ \ \mathbf{w}\cdot\Phi(\mathbf{m}_i) \ge \mathbf{w}\cdot\Phi(\mathbf{m}'_i) + 1. \qquad (20)\end{aligned}$$

Note that this reformulation of the problem does not affect its satisfiability, and that the solution of (20) is unique, since it is a strictly convex quadratic program.

In the second step we consider the Wolfe dual (Bertsekas et al. 2003) of (20):

$$\begin{aligned}&\text{Maximize } (w.r.t.\ \alpha):\\ &\sum_{i=1}^{k}\alpha_i - \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{k}\alpha_i\alpha_j\left(\left(\Phi(\mathbf{m}_i) - \Phi(\mathbf{m}'_i)\right)\cdot\left(\Phi(\mathbf{m}_j) - \Phi(\mathbf{m}'_j)\right)\right) \qquad (21)\\ &\text{subject to: } \alpha \ge \mathbf{0}\end{aligned}$$

This is a standard technique frequently used in the context of SVMs (Vapnik 1998; Herbrich et al. 2000). The Wolfe dual as in (21) has the same optimum value as the primal in (20). From the parameter vector α^* that solves the dual one can derive the solution $\mathbf{w}^*$ of the primal as

$$\mathbf{w}^* = \sum_{i=1}^{k} \alpha_i^* (\Phi(\mathbf{m}_i) - \Phi(\mathbf{m}_i')).$$

The third and final step is based on the observation that the dual (21) can be expressed in terms of inner products in the high-dimensional feature space. It is easy to verify that the following optimization problem is equivalent to (21):

$$\begin{aligned} &\text{Maximize } (w.r.t.\ \alpha): \\ &\sum_{i=1}^{k} \alpha_i - \frac{1}{2} \sum_{i=1}^{k} \sum_{j=1}^{k} \alpha_i \alpha_j \Big(\Phi(\mathbf{m}_i) \cdot \Phi(\mathbf{m}_j) - \Phi(\mathbf{m}_i) \cdot \Phi(\mathbf{m}_j') - \Phi(\mathbf{m}_i') \cdot \Phi(\mathbf{m}_j) \\ &\qquad + \Phi(\mathbf{m}_i') \cdot \Phi(\mathbf{m}_j') \Big) \\ &\text{subject to: } \alpha \geq \mathbf{0} \end{aligned} \tag{22}$$

For many kinds of mappings Φ, inner products can be computed efficiently using a RKHS kernel (see Vapnik 1998), even if Φ maps into a high-dimensional space. Our task, thus, is to find such a kernel for the specific mapping Φ that we use in our construction in (5–6).

Let us define an injective representation of partial assignments $\mathbf{m}$ from $\mathcal{X}^+$ by projecting them to *indicator vectors* $\mathbb{m} \in \mathbb{R}^{2n}$. Each attribute value is mapped onto a single dimension. If an attribute value is present in $\mathbf{m}$, the corresponding component of $\mathbb{m}$ is 1, otherwise 0. If an attribute is unspecified by $\mathbf{m}$, all corresponding components of $\mathbb{m}$ are set to 0. For example, if $\mathbf{X} = \{X_1, X_2, X_3, X_4\}$ and $\mathbf{m} = x_1 \overline{x_2} x_3$, then

$$\mathbb{m} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{matrix} x_1 \\ \overline{x_1} \\ x_2 \\ \overline{x_2} \\ x_3 \\ \overline{x_3} \\ x_4 \\ \overline{x_4} \end{matrix}$$

The indicator vectors $\mathbb{m} \in \mathbb{R}^{2n}$ for the partial assignments $\mathbf{m}$ to $\mathbf{X}$ (appearing as the local models of the formulae compared by the user) are similarly defined; if an attribute is unspecified by $\mathbf{m} \in \mathcal{X}^+$, all corresponding components of $\mathbb{m}$ are set to 0. Using this construction, inner products for an (effectively equivalent) variant Φ_λ of our mapping Φ can be computed as follows.

Theorem 2 *For the mapping* $\Phi_\lambda : \mathcal{X}^+ \mapsto \mathcal{F} = \mathbb{R}^{4^n}$

$$\Phi_\lambda(\mathbf{m})[i] = \begin{cases} \sqrt{c_\lambda(|\mathsf{val}(\mathsf{f}_i)|)}, & \mathsf{val}(\mathsf{f}_i) \subseteq \mathbf{m} \\ 0, & \text{otherwise,} \end{cases} \tag{23}$$

where

$$c_\lambda(k) = \sum_{l=k}^{n} \lambda_l \sum_{\substack{l_1 \geq 1, \ldots, l_k \geq 1 \\ l_1 + \cdots + l_k = l}} \frac{l!}{l_1! \ldots l_k!}, \tag{24}$$

and any $\mathbf{m}, \mathbf{m}' \in \mathcal{X}^+$ *and* $\lambda_1, \ldots, \lambda_n \geq 0$*, the kernel*

$$K(\mathbf{m}, \mathbf{m}') = \sum_{l=1}^{n} \lambda_l (\square\mathbf{m} \cdot \square\mathbf{m}')^l \tag{25}$$

computes the inner product $\Phi_\lambda(\mathbf{m}) \cdot \Phi_\lambda(\mathbf{m}') = K(\mathbf{m}, \mathbf{m}')$.

Proof The following chain of equalities holds.

$$\begin{aligned}
K(\mathbf{m}, \mathbf{m}') &= \sum_{l=1}^{n} \lambda_l (\square\mathbf{m} \cdot \square\mathbf{m}')^l \\
&= \sum_{l=1}^{n} \lambda_l \sum_{(i_1, \ldots, i_l) \in \{1, \ldots, 2n\}^l} (\square\mathbf{m}_{i_1} \square\mathbf{m}'_{i_1} \square\mathbf{m}_{i_2} \square\mathbf{m}'_{i_2} \ldots \square\mathbf{m}_{i_l} \square\mathbf{m}'_{i_l}) \\
&= \sum_{l=1}^{n} \lambda_l \sum_{(i_1, \ldots, i_l) \in \{1, \ldots, 2n\}^l} (\square\mathbf{m}_{i_1} \square\mathbf{m}_{i_2} \ldots \square\mathbf{m}_{i_l})(\square\mathbf{m}'_{i_1} \square\mathbf{m}'_{i_2} \ldots \square\mathbf{m}'_{i_l}) \\
&= \sum_{k=1}^{n} c_\lambda(k) \sum_{\{i_1, \ldots, i_k\} \subseteq \{1, \ldots, 2n\}} (\square\mathbf{m}_{i_1} \square\mathbf{m}_{i_2} \ldots \square\mathbf{m}_{i_k})(\square\mathbf{m}'_{i_1} \square\mathbf{m}'_{i_2} \ldots \square\mathbf{m}'_{i_k}) \\
&= \Phi_\lambda(\mathbf{m}) \cdot \Phi_\lambda(\mathbf{m}')
\end{aligned}$$

$c_\lambda(k)$ is the multiplicity with which a monomial of size k occurs. The multiplicity is influenced by two factors. First, different orderings of the index sequence $(i_1, \ldots, i_l)$ lead to the same term. This is counted by the multinomial coefficient $\frac{l!}{l_1! \ldots l_k!}$, where $l_1, \ldots, l_k$ are the powers of each factor. Second, all positive powers of any $\square\mathbf{m}_i \square\mathbf{m}'_i$ are equal. We therefore sum over all such equivalent terms

$$\sum_{\substack{l_1 \geq 1, \ldots, l_k \geq 1 \\ l_1 + \cdots + l_k = l}} \frac{l!}{l_1! \ldots l_k!}.$$

Note that many of the monomials always evaluate to zero under our encoding $\square\mathbf{m}$ of $\mathbf{m}$. Specifically, monomials corresponding to expressions $(x_i \wedge \overline{x_i} \wedge \cdots)$ will always be nullified.[6] In particular, it is therefore sufficient to consider only monomials of size less or equal to n, since all others will always evaluate to zero. □

The kernel in (25), which is similar to a polynomial kernel (Vapnik 1998), allows us to compute inner products in the high-dimensional space in linear time, and, for strictly positive $\lambda_1, \ldots, \lambda_n$, moving from Φ to Φ_λ does not change the satisfiability of

[6] This clarifies our declaration in Sect. 2.1 that we effectively use only the $(3^n - 1)$ dimensions of $\mathcal{F}$, corresponding to the non-empty partial assignments to $\mathbf{X}$. However, keeping the computations in the 4^n-dimensional space $\mathcal{F}$ allows us to use the machinery of the RKHS kernels.

our constraint system $\mathfrak{C}$. To see the latter, observe that any solution $\mathbf{w}_\lambda$ of (19) for Φ_λ corresponds to a solution $\mathbf{w}$ of (19) for Φ via

$$\mathbf{w}_\lambda[i] = \frac{\mathbf{w}[i]}{\sqrt{c_\lambda(|\mathsf{val}(\mathfrak{f}_i)|)}}.$$

The only difference between the mappings Φ_λ and Φ is that the former biases the inference's prior towards smaller size monomials, "preferring" more general explanations for user preference statements. To a large degree, this bias can be controlled via the kernel parameters $\lambda_1, \ldots, \lambda_n$.

Now, using the kernel inside of the dual leads to the following equivalent optimization problem.

$$\begin{aligned}&\text{Maximize } (w.r.t.\ \alpha):\\ &\sum_{i=1}^{k}\alpha_i - \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{k}\alpha_i\alpha_j\left(K(\mathbf{m}_i,\mathbf{m}_j) - K(\mathbf{m}_i,\mathbf{m}'_j) - K(\mathbf{m}'_i,\mathbf{m}_j) + K(\mathbf{m}'_i,\mathbf{m}'_j)\right) \quad (26)\\ &\text{subject to: } \alpha \geq \mathbf{0}\end{aligned}$$

It is known that such convex quadratic programs can be solved in polynomial time (Bertsekas et al. 2003). To compute the value of $\mathfrak{U}$ for a given alternative $\mathbf{x} \in \mathcal{X}$, it is sufficient to know only the dual solution and the kernel:

$$\mathfrak{U}(\Phi(\mathbf{x})) = \mathbf{w}^* \cdot \Phi_\lambda(\mathbf{x}) = \sum_{i=1}^{k}\alpha_i\left(K(\mathbf{m}_i,\mathbf{x}) - K(\mathbf{m}'_i,\mathbf{x})\right) \quad (27)$$

Hence, *neither computing the solution* $\mathfrak{U}$ *of the constraint system* $\mathfrak{C}$*, nor computing the values of* $\mathfrak{U}$ *on* $\mathcal{X}$ *requires any explicit computations in* $\mathbb{R}^{4^n}$. Through the use of kernels, all computations can be done efficiently in the low-dimensional input space. We have extended SVM$^{\text{light}}$ to solve this type of quadratic optimization problem.[7] It can efficiently handle large-scale problems with $n, m \approx 10,000$.

As a final comment on the mechanics of our inference procedure, note that it would be unreasonable to expect that preference statements of a user will always be consistent. In case of an inconsistent preference expression, the set of constraints in the primal optimization problem (20) becomes infeasible and the solution of the dual optimization problem (21) is unbounded. Fortunately, the soft-margin technique (Cortes and Vapnik 1995) can be used to handle such inconsistent preference specifications. The idea is to introduce non-negative slack variables ξ_i into the constraints and penalize their sum in the objective function. This leads to the following modified primal optimization problem:

$$\begin{aligned}&\text{Minimize } (w.r.t.\ \mathbf{w}, \xi_1, \ldots, \xi_k): \frac{1}{2}\mathbf{w}\cdot\mathbf{w} + C\sum_{i=1}^{k}\xi_i\\ &\text{subject to: } \mathbf{w}\cdot\Phi(\mathbf{m}_1) \geq \mathbf{w}\cdot\Phi(\mathbf{m}'_1) + 1 - \xi_1\\ &\qquad\ldots\\ &\qquad\mathbf{w}\cdot\Phi(\mathbf{m}_k) \geq \mathbf{w}\cdot\Phi(\mathbf{m}'_k) + 1 - \xi_k\\ &\qquad\xi_1 \geq 0, \ldots, \xi_k \geq 0\end{aligned} \quad (28)$$

[7] SVM$^{\text{light}}$ is available at `http://svmlight.joachims.org/`

The parameter C in the objective allows trading-off constraint violations against margin size. Note that the sum of the slack variables is an upper bound on the number of constraints that are violated by the solution $\mathbf{w}$.

The dual of this optimization problem is similar to the original dual, except that the dual variables α_i are now upper-bounded by C.

$$\begin{aligned}&\text{Maximize } (w.r.t.\ \alpha):\\ &\sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{k}\alpha_i\alpha_j((\Phi(\mathbf{m}_i)-\Phi(\mathbf{m}'_i))\cdot(\Phi(\mathbf{m}_j)-\Phi(\mathbf{m}'_j))) \qquad (29)\\ &\text{subject to: } \mathbf{C}\geq\alpha\geq\mathbf{0}\end{aligned}$$

The properties of the dual solution discussed earlier for the consistent case also hold for the soft-margin formulation.

4 Inference semantics

Since the user's statements typically provide only partial information about her preferences, the constraint system in (19) is underconstrained, and thus the utility revelation takes the form of inductive[8] reasoning. If the system has access to a prior $Pr(\mathfrak{U})$ over utility functions, a reasonable inductive inference procedure would be to pick the most likely utility function $\mathfrak{U}$ that fulfills all constraints. In particular, for the Gaussian prior $Pr(\mathfrak{U}) \sim e^{-||\mathbf{w}||^2}$ this procedure results in finding the weight vector $\mathbf{w}$ with minimum L_2-norm that fulfills the constraints. This is exactly our objective in (20).

To illustrate the behavior arising from this prior, consider the set of three statements:

$$\begin{aligned}s_1 &= (X_1 \vee X_2) \succ (\neg X_3),\\ s_2 &= (X_3) \succ (X_4),\\ s_3 &= (X_1) \succ (X_2).\end{aligned}$$

For the small set of constraints induced by these statements, we can compute the solution without the use of kernels and get the following weights.

$$\begin{array}{llll} w_{x_1}=0.75 & w_{x_2}=-0.25 & w_{x_3}=0.5 & w_{x_4}=-0.5\\ w_{\overline{x_1}}=0.4 & w_{\overline{x_2}}=0 & w_{\overline{x_3}}=-0.45 & w_{\overline{x_4}}=0\\ w_{x_1x_2}=0.05 & w_{\overline{x_1}x_2}=0.4 & & \end{array}$$

All other weights are set to zero. Below is an illustrative excerpt of the ordering induced by the utility function generated in our framework:

$$\begin{aligned}\mathfrak{U}(\Phi(x_1\overline{x_2}x_3\overline{x_4})) &= 1.25\\ \mathfrak{U}(\Phi(x_1x_2x_3\overline{x_4})) &= 1.05\\ \mathfrak{U}(\Phi(\overline{x_1x_2}x_3\overline{x_4})) &= 0.9\\ \mathfrak{U}(\Phi(x_1x_2x_3x_4)) &= 0.55\end{aligned}$$

[8] Actually, the preference elicitation problem is more accurately modeled as a transductive inference problem (see e.g., Vapnik 1998), but we do not consider this refinement at this time for simplicity.

$$\mathfrak{U}(\Phi(x_1x_2\overline{x_3x_4})) = 0.1$$
$$\mathfrak{U}(\Phi(x_1x_2\overline{x_3}x_4)) = -0.4$$
$$\mathfrak{U}(\Phi(\overline{x_1x_2x_3}x_4)) = -0.55.$$

We believe that this ordering reflects a natural interpretation of the statements. Furthermore, alternatives for which the statements give no clear judgment receive utility values closer to zero than those for which a statement clearly applies. In general, the Gaussian prior appears reasonable in situations where we expect the utility function to have a compact form (that is, most weights in $\mathbf{w}$ are small).

Now let us go back to the semantics of the parameters of $\mathfrak{U}$. Recall that (a) the purpose of each parameter w_i is to capture the marginal value of the event $\mathsf{val}(\mathfrak{f}_i)$, and (b) we strive to a least committing interpretation of preference expressions. Despite the fact that we care only about the ordinal properties of $\mathfrak{U}$ (and thus these functions are isomorphic under any order-preserving transformation), our machinery implicitly provides us with a reference point $\mathbf{0} \in \mathbb{R}^{2^n}$. Specifically, we will have $w_i = 0$ both in case the user is known to associate no value with $\mathsf{val}(\mathfrak{f}_i)$, and in case where this is not known, but we have no reason to believe otherwise.

Given an alternative $\mathbf{x} \in \mathcal{X}$ and a utility function $\mathfrak{U}$ consistent with the constraints posed by the user's preference statements, suppose that we have:

$$\mathfrak{U}(\Phi(\mathbf{x})) = 0 \tag{30}$$

Such an evaluation of $\mathbf{x}$ may correspond to two semantically different situations. If for all $\mathfrak{f}_i$ such that $\mathsf{val}(\mathfrak{f}_i) \subseteq \mathbf{x}$, we have $w_i = 0$, then the user is considered[9] to be *indifferent* about $\mathbf{x}$. Informally, the user is indifferent about an alternative if she has neither reasons to like it nor reasons to dislike it. Alternatively, if for some $\mathfrak{f}_i$ such that $\mathsf{val}(\mathfrak{f}_i) \subseteq \mathbf{x}$, we have $w_i \neq 0$ [and yet we have $\mathfrak{U}(\Phi(\mathbf{x})) = 0$], then the user is considered to be *neutral* about $\mathbf{x}$. The user is considered to be *neutral* about an alternative if her reasons to like it somehow "balance" her reasons to dislike it.

To summarize the above, we have two semantically different situations that correspond to (30), and there is no way to distinguish between them from the value of $\mathfrak{U}$ alone. But should we? First, from the practical point of view of providing the user with a choice-guiding procedure, so far we have seen no benefits for distinguishing between neutrality and indifference. Interestingly, this position also underlies one of the best-known approaches to the logic of monadic preference concepts, namely that of Chisholm and Sosa (1966). In their terminology, to be neutral means to be equal in value to something that is indifferent, which is exactly the connection between these two cases in our framework.

The representation of neutrality and indifference in our framework bring us to discuss the monadic statements of preference. While, as we already mentioned, the actual value of the reference point has no specific semantics, the fact that our machinery has a reference point independent of the given set of constraints turns out to be very helpful. In philosophical logic there are several traditions for interpreting concepts such as "good" and "bad" (for a comparative survey see Hansson 2001b, p. 118), but probably the most natural one for our least committing methodology seems to be (again) the approach of Chisholm and Sosa (1966). According to these authors, the meaning of "good" and "bad" should be defined with respect to neutrality, and, together with the

[9] It is probably better to say that "preference elicitation provides us with no reasons to believe that the user is not indifferent about $\mathbf{x}$," but we allow ourselves to use a shorter formulation.

aforementioned connection between indifference and neutrality, "a state of affairs[10] is good provided it is better than some state of affairs that is indifferent, and ... a state of affairs is bad provided some state of affairs that is indifferent is better than it" (Chisholm and Sosa 1966). This interpretation gives rise to an extremely straightforward encoding of monadic preference statements in our framework. Specifically, a statement "φ is good" is translated into a set of $|M(\varphi)|$ constraints:

$$\sum_{\mathsf{f}_i:\mathsf{val}(\mathsf{f}_i)\in 2^{m(\varphi)}} w_i \;\; > 0, \tag{31}$$

which can be seen as a special case of the constraint form as in (12).

Finally, let us now consider higher order preference statements, such as "φ is preferred to ψ more than ρ is preferred to ω." It is not hard to see that, following the principles of our interpretation, such a statement should be encoded as:

$$\sum_{\mathsf{f}_i:\mathsf{val}(\mathsf{f}_i)\in 2^{\boldsymbol{m}(\varphi)}} w_i - \sum_{\mathsf{f}_j:\mathsf{val}(\mathsf{f}_j)\in 2^{\boldsymbol{m}(\psi)}} w_j \;\; > \sum_{\mathsf{f}_k:\mathsf{val}(\mathsf{f}_k)\in 2^{\boldsymbol{m}(\rho)}} w_k - \sum_{\mathsf{f}_l:\mathsf{val}(\mathsf{f}_l)\in 2^{\boldsymbol{m}(\omega)}} w_l \tag{32}$$

and the description complexity of such constraints is of the same order as those in (12).

5 Experiments

While the proposed approach is appealing from a theoretical perspective, it remains to be shown whether its assumptions match the properties of real-world data. In the following, we evaluate the effectiveness of our approach on turning a set of preference statements S into a desirable ordering, as well as its empirical computational efficiency. In particular, we address whether the method we propose can

1. generate an accurate ordering from instance comparisons?
2. generate an accurate ordering from generalizing statements?
3. exploit instance comparisons to retrieve some best k items in the collection?
4. exploit generalizing statements to retrieve some best k items in the collection?

The last two questions address the applicability of our approach in systems focusing on *one-shot* preference queries that are used only within a single search session of the user. For instance, such queries are typical to "pull-based" e-commerce retail systems in which the users are not likely to have repeated interests in the very same category of products. In contrast, the first two questions address the applicability of our approach in "push-based" systems that aim at supporting *standing* preference queries of the users, describing long-term information needs that should be repeatedly addressed as the content of the catalog is changing.

While important, we do not include user interface issues into our evaluation, but focus the evaluation on the inference method itself. Designing interfaces that let users easily provide statements via explicit entry, ratings, or implicit signals like clicks and dwell times are beyond the scope of this paper and are the subject of future work.

In our experiments we use two datasets, namely the EachMovie dataset (McJones 1997) and the MovieLens (1 Million) collection (Riedl et al. 2006). Both datasets consist of user ratings for movies, but contain no generalizing statements. To our

[10] In our terminology, "state of affairs" corresponds to alternative.

knowledge, all (publicly available) preference-related data sets contain only ratings or instance comparisons (of books, movies, etc.), but unfortunately no generalizing statements. While it is straightforward to generate instance comparisons from ratings, we will use a rule-learning algorithms to extract generalizing statements from ratings as described below.

The *EachMovie* dataset (collected and provided by P. McJones at DEC/Compaq Research) consists of six-point-scale movie ratings (i.e., 0–5 stars) collected from 72,916 users on a corpus of 1,628 movies. Each movie is described by a set of attributes, out of which we use twelve attributes corresponding to the decade of the movie, whether it is currently in the movie theaters, and a binary classification according to ten (non-disjoint) genre categories. We present results that are averaged over the 50 users with the largest number of ratings. As the set of alternatives for each user we randomly select 500 movies among those rated by this user.

The *MovieLens* dataset (collected and provided by the GroupLens Research Project at the University of Minnesota) consists of 6,040 users rating 3,900 movies on a five point scale (i.e., 1–5 stars). Movies are described using the decade of its release and 18 (non-disjoint) genre categories. Again, we use the 50 most prolific users and select 500 rated movies for each. Unlike for the EachMovie dataset, these movies are not selected randomly from the set of rated movies, but with the goal of making this dataset less "easy" (i.e., lower the average rating, which is almost 3.5 of 5 stars). We therefore selected the 50 highest rated movies and the 450 lowest rated movies for each user.

In our experiments we generate one separate ordinal utility function for each user, and then report average results over all users. Note that the problem is less of a prediction problem, but more of a communication and decoding problem—the set of statements S is a message that is interpreted by our method to recover the desired ordering. The smaller S and the more accurate the ordering, the more effective is our method. Furthermore, this inference problem is transductive (Vapnik 1998), not inductive. In contrast to inductive learning where evaluation is done on a randomly drawn test set, in transductive learning there is a fixed set of partially labeled examples (i.e., the 500 movies) which is known to the algorithm at inference time.

Finally, all our experiments reported here use a soft-margin parameter $C = \frac{1}{k}$ (see 28), where k is the number of constraints in the constraint set from (18). We found these results to be representative for a large range of different values of C.

5.1 Generating an ordering from instance comparisons

In this first experiment we explore in how far our method can recover an ordering given a set S of instance comparisons. We generate a separate ranking for each user. The instance comparisons for each user are generated by repeatedly selecting two movies with different ratings uniformly at random from the set of 500 movies. Each such pair forms one instance-level statements. In this setting, our method is equivalent to an ordinal regression SVM (Herbrich et al. 2000) with a particular feature encoding and sampling of statements. In practical applications, such instance comparisons can be derived from implicit feedback signals like clickthrough (see e.g., Joachims et al. 2005) or from explicit feedback like in conjoint analysis (see e.g., Green et al. 2001).

The accuracy of the orderings generated from S is measured in terms of ranking error, that is the fraction of times where the user rating and the utility function disagree on the ordering of two movies. This error measure considers only movie pairs

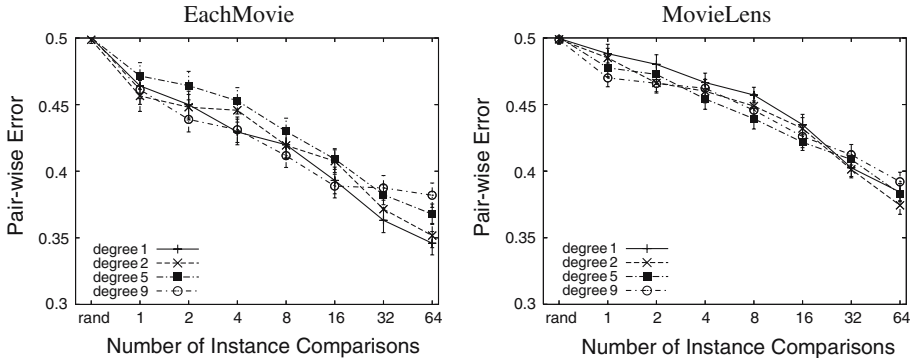

Fig. 1 Fraction of misordered pairs (i.e., ranking error) averaged over all users. The x-axis show the number of instance comparisons in S that were used to create the ranking. *Errorbars* show one standard error. ($C = \frac{1}{k}$)

rated unequally by the user. Ties in the ordering induced by the utility function are broken randomly. Note that random performance according to this error measure is a score of 0.5, and that a score of 0.0 indicates a perfect ordering. However, it is unlikely that a perfect ordering can be achieved given the rather coarse descriptions of the movies available in the datasets.

Figure 1 plots the number of statements versus the error of the generated ranking for the EachMovie data on the left, and the MovieLens data on the right. Each curve in Fig. 1 gives the performance for a different choice of kernel degree, that is, different choice of kernel parameters $\lambda_1, \ldots, \lambda_n$. The "degree" d indicates that all λ_i with $i > d$ are set to zero, while all others are one. This eliminates all monomials of size greater than d.

Overall, the method is able to generate a reasonable utility function that ranks items significantly and substantially better than the random baseline of 0.5. The results are using the default value of $C = \frac{1}{k}$ for the soft-margin parameter, where k is the number of constraints in the constraint set from (18). The method seems to be rather robust regarding the choice of C, since other values of C gave comparable results over a wide range. Regarding the choice of degree for the kernel, the method is robust as well. Most values gave reasonable accuracy and most differences are not significant. However, for the EachMovie data with large numbers of instance comparisons, lower degree kernels appear to have some advantage over higher degree kernels. A priori, the latter outcome may have two (possibly complementary) explanations:

1. Interactions between the data attributes do not significantly affect the preferences of a typical user, and thus her preferences can be effectively captured by a function linear in these attributes,
2. The small amount of information carried by each individual instance comparison *and* the small number of such statements in S do not allow us to separate between (preference-wise) important and unimportant interactions between the attributes.

However, while in principle both explanations are possible (together and separately), the results of our next set of experiments suggest that the first explanation is *very*

unlikely, further supporting the attractiveness of eliciting generalizing preference information.

5.2 Generating an ordering from generalizing statements

Generalizing statements have the potential to be more informative and flexible than instance comparisons, since they can be constructed without being limited to the (currently visible) items in the database. Analogous to the setup of the previous section, we now consider as input a set of generalizing statements S.

The main obstacle is that our (and all other publicly available) preference-related data sets do not contain generalizing statements. To overcome this obstacle, we designed an experimental setup in which generalizing statements are generated from ratings-based instance comparisons. To simulate generalizing preference statements, we generate such statements using the C4.5 decision trees learning algorithm (Quinlan 1993) on the following binary classification problem. As training examples, we consider all 500×500 pairs (A, B) of movies by concatenating their attribute vectors. For each user we generate a separate training set. If the user rated the first movie higher (lower) than the second movie, the pair is labeled positive (negative). No pair is generated if both movies have the same rating because it was unclear how to translate such cases into training examples for the classification task. On this data, we run the C4.5 decision tree learner.[11] Using the `c45rules` software included in the C4.5 package we then convert the resulting decision tree into a set of rules ordered by their level of confidence, and interpret each of the extracted rules as a single generalizing preference statement. For example, the highest ranked rule for the user that rated the largest number of movies in the EachMovie dataset was the rule (a) below.

(a)
```
B_decade = 1990s
B_Art_Foreign = 1
B_Family = 0
B_Romance = 0
  -> A preferred to B
```

(b)
```
A_decade = 1980s
A_Thriller = 1
B_Classic = 0
B_Horror = 1
  -> A preferred to B
```

This rule can be interpreted as the monadic preference statement "the user does not like foreign films from the 1990s that are not Romance or Family movies." For the same user, the highest ranked dyadic rule is rule (b), meaning "the user prefers thrillers from the 1980s over non-classic horror movies."

The set of rules (or equivalently the corresponding generalizing preference statements) generated by C4.5 can be thought of as a compact description of the user's desired ordering, as expressed in the user's ratings. This simulates a communication task as elaborated above, where a user prefers communicating her preference ordering to the system by providing a set of generalizing statements instead of explicitly rating the individual movies.

Figure 2 shows how well the utility function orders the movies depending on the number of generalizing preference statements used to generate this function. The form of the plot is the same as in the previous section, but instead of instance-level statements we use the top t generalizing preference statements as returned by `c45rules`. Each curve in Fig. 2 gives the performance for a different choice of kernel degree. As expected, the error rate decreases as more preference statements are provided. For small numbers of preference statements, all degrees perform roughly equivalently.

[11] `http://www.cse.unsw.edu.au/~quinlan/`

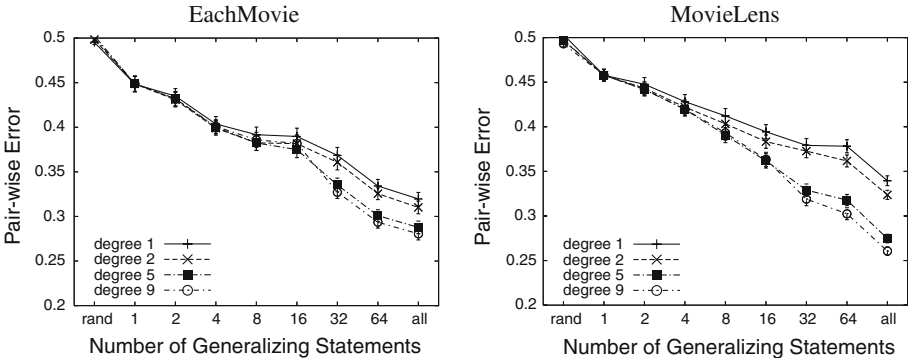

Fig. 2 Fraction of misordered pairs (i.e., ranking error) averaged over all users. The *x*-axis show the number of statements in *S* that were used to create the ranking. *Errorbars* show one standard error. ($C = \frac{1}{k}$)

For larger sets of preference statements, high-degree kernels substantially and significantly outperform low-degree kernels. This finding is largely independent of the choice of regularization parameter C and is not limited to our default choice of $C = \frac{1}{k}$. In further experiments we found that no value of C makes low-degree kernels perform as good as high-degree kernels. The best results were obtained for degrees greater or equal to 5. Increasing the degree beyond 5 gave only small further improvements and by degree 9 the performance seems to asymptote. It appears that low-degree kernels cannot capture the dependencies explicitly expressed in the preference statements, and thus the ability to handle large-degree monomials (i.e., non-linear interactions between attributes $\mathbf{X}$) is beneficial. Unlike for the instance comparisons discussed in the previous section, the generalizing statements focus on the relevant high-degree monomials so that overfitting is avoided even for high-degree kernels. Furthermore, the error rate achieved with the generalizing statements is lower than when using the instance comparisons like in the previous section.

Since, we are using a very coarse description of the movies, the attributes do not suffice to produce a perfect ordering from a small number of preference statements. In particular, the average error rate of the complete set of C4.5 rules is 0.24 for the EachMovie data and 0.16 for the MovieLens data. Note that this pairwise classification performed by C4.5 is potentially easier than the utility revelation problem, since the rules do not have to form an ordering. Comparing the C4.5 performance against the error rates of around 0.28 and 0.26 achieved by the ordinal utility function for the high-degree kernels, we conclude that our method performs the translation into a consistent ordering effectively and with good accuracy.

5.3 Using instance comparisons to find the best alternatives

The two experiments above verify that our preference revelation method is indeed well-founded in how it interprets both instance and generalizing preference statements. However, the experiment design above addresses mostly applications that should continuously push information to the user based on its long-term user model, and less reflects the needs of search-engine like applications that aim to respond to some one-shot user need. We believe that in the latter case the system does not need

to discover the complete preference ordering of the user, but only needs to identify a small set of highest-utility alternatives among those currently available in the catalog. For example, suppose that a user of a movie recommendation system is not interested in subscribing to periodic personalized recommendations, but only wants to get a single recommendation "for today." In such case it is important to quickly identify movies that the user would have found most interesting, and it is not important whether less interesting (to this user) movies in the catalog are ordered correctly among themselves.

To evaluate our system in this type of setting, we conducted a set of experiments that simulate how users might provide instance comparisons in an interactive system. Each experiment for each user goes through a series of iterations. Starting with a random ordering, the user marks the best movies among the top $t = 10$ movies that are presented to her. The system then generates an instance comparison between each marked and each unmarked movie in this top-t set, adds these statements to S, generates a new ranking, and repeats. In our experiments, we simulate the user feedback by marking all movies which have the highest number of stars among the top t. Performance is measured by the average star-rating among the t highest ranked movies.

Figure 3 shows the average number of stars among the ten movies ranked highest by the SVM for up to nine iterations of feedback. The curves for EachMovie and MovieLens show similar behavior. The first iterations lead to bigger performance gains than later iterations. Like in the previous experiments with instance comparisons, there seems to be a trend that the performance for lower degree kernels is better than that for higher degree kernels. However, the difference is small and in many cases not significant.

5.4 Using generalizing statements to find the best alternatives

One possible problem with the instance comparisons (as used in the previous experiment) is that they might miss "clusters" of movies that the user rates highly. Consider, for example, a user that likes Western and Romance movies. If the initial ranking

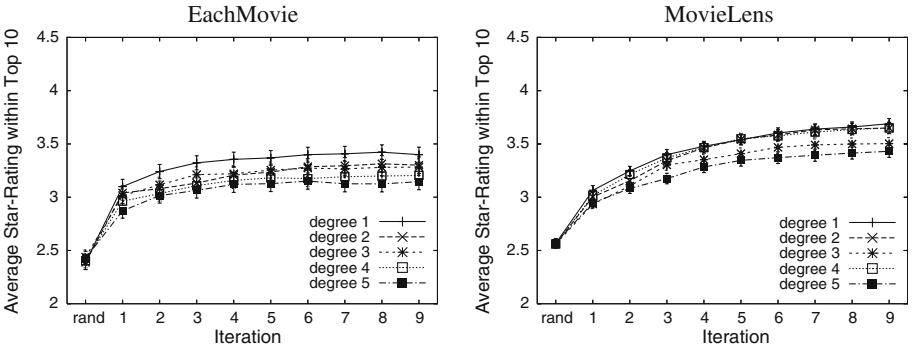

Fig. 3 Average star rating of movies ranked in the top 10 averaged over all users. The x-axis show the number of instance comparisons in S that were used to create the ranking. *Errorbars* show one standard error. ($C = \frac{1}{k}$)

contains a Western, but no Romance movies, the instance comparisons generated from the initial ranking make it unlikely that a Romance movie will ever enter the top t in any further iteration. This means that the user will never be able to state preferences for Romance movies. Generalizing statements can remedy this problem, since the user can directly express her interests.

As before, we generate generalizing statements using C4.5. To focus on statements describing the top of the ranking, we use all movies with five stars (or at least the ten highest rated movies) among the 500 movies of each user as positive examples, and all movies with a below average star-rating as negative examples. On this focused dataset we then run C4.5 and C4.5 rules and extract the resulting monadic generalizing statements from the rules. On average, this generated 4.0 statements for the EachMovie data and 6.4 statements for the MovieLens data.

Figure 4 shows the average star rating among the top ten movies after processing a given number of generalizing statements. Again, the behavior of the method is similar for both corpora. With just a single generalizing statement, the performance jumps substantially and significantly above the random baseline. Adding further statements provides smaller gains. It appears that the higher degree kernels provide a small benefit over a linear utility function, although the difference is not significant.

The "dot" in either plot of Fig. 4 marks the performance achieved by directly using the C4.5 rules and ranking movies by the confidence factors that C4.5 estimates. Interestingly, the utility function generated from the rules is more accurate than the rules themselves. We conjecture that this is due to the fact that C4.5 rules are intended for classification, not ranking, and that the C4.5 method for estimating confidence factors might be suboptimal. In any case, the graph shows that our SVM method can successfully extract ranking information from the generalizing statements to find the highest rated items in the database.

Comparing the performance in Fig. 4 against the experiments with instance comparisons in Fig. 3, we see that the generalizing statements again lead to a better performance than the instance comparisons. In practice, we conjecture that a combination of instance comparisons and generalizing statements might provide an optimum of usability and performance.

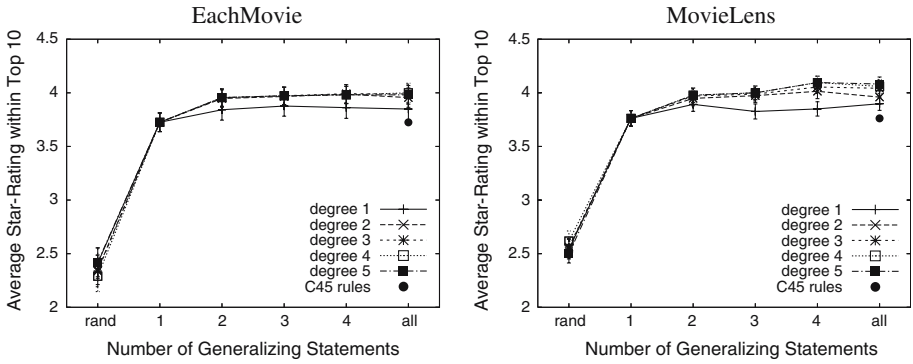

Fig. 4 Average star rating of movies ranked in the top ten averaged over all users. The x-axis show the number of statements in S that were used to create the ranking. *Errorbars* show one standard error. ($C = \frac{1}{k}$)

5.5 Computational efficiency

Regarding computational efficiency, the average CPU-time of SVM^{light} for solving the quadratic program for a set of 64 generalizing statements was less than 0.1 s. In other experiments we have found that SVM^{light} can efficiently handle much larger scale problems with $n \approx 10,000$ attributes and $m \approx 10,000$ statements.

6 Related work

Modeling user preferences and reasoning about them has became an active topic of research in computer science over the last three decades. Research on preferences, however, has a much older history. As the paradigm of human decision making, preferences are a key concept in decision theory, philosophy, psychology, and economics (e.g., Green et al. 2001; Hansson 2001b; Keeney and Raiffa 1976). The central research goals in these areas have been to provide logical, cognitive, and mathematical models of human decision making. The main additional value that computer science has brought into the research on user preferences is an attempt to automate the whole loop of preference information processing, modeling, and decision support. The goal of such automation is to make (both previously developed and new) logical and mathematical foundations usable in systems where both the number of choice alternatives is huge, and no professional decision analyst is available to help each user in her task.

Since the paradigm of human preferences have been extensively studied across this many fields, it is simply impossible in one paper to relate to every significant proposal in this area, or even just provide a comprehensive survey of this multi-disciplinary effort. Therefore, below we describe only previous work that is most closely related to our approach.

The key properties of our approach (apart from its tractability) are (a) the structure and semantics of the space in which we decompose the ordinal preferences of the users, and (b) the fact that this decomposition is linear, making the technique completely non-parametric. The search for the "right" dimensional structure for preference decomposition has been always one of the key research issues in the area. In the recent survey article on representation and reasoning about preferences (Doyle 2004), Doyle writes: "…we seek attributions that cleanly divide up outcomes (alternatives) into attributes that correspond to distinct dimensions of variation of utility. That is we seek coordinate systems over outcomes such that utility takes a simple form in the chosen coordinate system." Unfortunately, it has been well known that dimensions natural to human decision analysts (and definitely some dimensions *given* to them) typically provide at most partial decomposition of the preferences, reducing the utility function to a weighted sum of some sub utility functions over subsets of these dimensions (Keeney and Raiffa 1976). Thus, while listing open problems in the area, Doyle writes: "Can one recast the underlying set (of attributed outcomes) in terms of a different span of dimensions such that the utility function becomes linear? If so, can one find new linearizing dimensions that also mean something to human interpreters?"

To the best of our knowledge, so far this question has not been answered. In fact, we found only one work directly attempting to shed some light on this issue, namely the work of Shoham on utility distributions (Shoham 1997a, b). In an attempt to provide a unified view on probabilities and utilities, Shoham showed that, in principle, such a

set of linearizing dimensions (called, in Shoham 1997a, b, factors) exist for *any* utility function, and that this set of dimensions may have to be exponentially larger than the original set of attributes. However, the result of Shoham is more foundational than operational. The connection between the attributes and the particular set of factors proposed in (Shoham 1997a, b) is not generally natural, and thus it is rather unclear how to perform preference elicitation with respect to this set of factors. Likewise, no efficient computational scheme for reasoning about this, potentially very large, set of factors has been proposed until these days. In some sense, in our work we extend the basic ideas and intuitions of Shoham, and, for the first time, provide an affirmative, practically usable, answer to the question of existence of a generic, linearizing space of dimensions.

In principle, "unstructuring" preferences by considering them in a linearizing set of dimensions is not the only way to achieve a non-parametric framework for modeling user preferences. One can design an inference mechanism that takes a set of observations on user preferences (e.g., user's own statements), and provides us with a function over the original set of attributes, such that the choice of function is optimal with respect to certain desired criteria (e.g., structural compactness). Clearly, these criteria should be reasonable enough to provide an axiomatic basis for the inference, and only recently such a set of criteria was proposed by Herfert and La Mura (Herfert and La Mura 2004). The inference method proposed in (Herfert and La Mura 2004) is based on the notion of decision-theoretic entropy (La Mura 1999, 2003), and it returns a unique utility function that is minimally committing with respect to missing information and individual judgments. However, the computational mechanism behind the methodology of Herfert and La Mura constitutes an explicit, non-linear optimization in the functional space $\mathbb{R}^{2^n}$, where n is the number of original attributes. Moreover, the description of the inferred utility function can be arbitrarily complicated, since the "right" utility function can still be poorly structured with respect to the attribute space $\mathcal{X}$. Therefore, while providing a powerful alternative to the conjoint analysis family of methods, the inference method in (Herfert and La Mura 2004) is not generally tractable. Having said that, we believe that there is a fundamental connection between the inference mechanism in our methodology, and the one proposed by Herfert and La Mura, and currently we investigate this issue.

As pointed out during the paper, our approach is closely connected to methods from machine learning and draws extensively upon techniques developed there. In particular, we translate the preference elicitation problem into a Support Vector Machine (Vapnik 1998; Cortes and Vapnik 1995) formulation, in which it is possible to use RKHS Kernels (Kimeldorf and Wahba 1971; Wahba 1990; Boser et al. 1992; Vapnik 1998) to efficiently work with high-dimensional vectors. For such formulations, it is known that the solution can always be expressed as a linear combination of the training data (Kimeldorf and Wahba 1971; Wahba 1990). In particular, our formulation builds upon the ordinal regression setup in Herbrich et al. (2000). However, in contrast to this and other methods for learning with preferences (see, e.g., Cohen et al. 1999; Freund et al. 2003), our approach explicitly addresses the semantics of translating logical preference expressions to ordinal utility functions, and by that allows us handling not only instance comparisons, but also generalizing preference statements. An interesting property of the preference elicitation problem as considered in our approach is that it is inherently transductive (see e.g., Vapnik 1998), not inductive as typically considered in machine learning. In transductive learning, the goal is to optimize performance on a finite and knows set of examples or alternatives

(e.g., items in a catalog), not over a distribution of examples like in inductive learning. Exploiting the transductive setup more directly as part of the inference mechanism is an interesting area for future work.

7 Conclusions

We have described a novel approach to ordinal utility revelation from a set of qualitative preference statements. The task is ubiquitous in the design of systems that adapt to users by modeling user preferences. To the best of our knowledge, our proposal constitutes the first solution to this problem that can handle heterogeneous preference statements both efficiently and effectively. The key technical contribution is a computationally tractable, non-parametric transformation into a space where ordinal utility functions decompose linearly and where dimensions have clear and intuitive semantics.

Our ongoing and future work builds upon the foundations laid in this paper in several directions. First, we would like to provide informative upper bounds on the number of preference statements that a user will have to specify before the inferred utility function approximates her preferences sufficiently well. Furthermore, we would like to perform a deeper analysis of the semantics of our inference procedure, connecting it, for instance, with the recent axiomatic approaches for preference revelation such as this in La Mura (1999, 2003).

Along with advances in theoretical foundations, we would like to explore applications in real-world settings, removing the assumptions about the structure of user preferences and removing restrictions on the form of statements currently imposed in existing operational systems. We believe that the framework proposed in this work pushes these limits of explicit user modeling, and hence provides an attractive infrastructure for user-centric electronic catalogs and product configuration systems.

Acknowledgements This work was funded in part under NSF CAREER Award IIS-0237381, and BSF Award 2004216 of United States-Israel Bi-national Science Foundation.

References

Bacchus, F., Grove, A.: Graphical models for preference and utility. In: UAI-1995, pp. 3–10. Montreal (1995)

Bertsekas, D., Nedic, A., Ozdaglar, A.: Convex Analysis and Optimization. Athena Scientific. Nashua, NH (2003)

Blythe, J.: Visual exploration and incremental utility elicitation. In: AAAI-02, pp. 526–532. Edmonton (2002)

Boser, B.E., Guyon, I.M., Vapnik, V.N.: A traininig algorithm for optimal margin classifiers. In: Haussler, D. (ed.) Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory, pp. 144–152. Pittsburg, PA (1992)

Boutilier, C.: Toward a logic for qualitative decision theory. In: Proceedings of the Third Conference on Knowledge Representation (KR–94), pp. 75–86. Bonn (1994)

Boutilier, C., Bacchus, F., Brafman, R.I.: UCP-networks: A directed graphical representation of conditional utilities. In: UAI-2001, pp. 56–64. Seatlle, WA (2001)

Boutilier, C., Brafman, R., Domshlak, C., Hoos, H., Poole, D.: CP-nets: A tool for representing and reasoning about conditional *ceteris paribus* preference statements. J. Artif. Intel. Res. **21**, 135–191 (2004)

Boutilier, C., Patrascu, R., Poupart, P., Schuurmans, D.: Regret-based utility elicitation in constraint-based decision problems. In: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence. Edinburgh, Scotland (2005)
Brafman, R., Domshlak, C., Kogan, T.: Compact value-function representations for qualitative preferences. In: UAI-04, Banff, Canada (2004)
Burke, R.D., Hammond, K.J., Young, B.C.: The FindMe approach to assisted browsing. IEEE Expert **12**(4), 32–40 (1997)
Chai, J., Horvath, V., Nicolov, N., Stys, M., Kambhatla, N., Zadrozny, W., Melville, P.: Natural language assistant: A dialog system for online product recommendation. AI Mag. **23**(2), 63–76 (2002)
Chisholm, R.M., Sosa, E.: On the logic of 'Intrinsically Better'. Am. Philos. Q. **3**, 244–249 (1966)
Chomicki, J. Preference formulas in relational queries. ACM Trans. Database Syst. **28**(4), 427–466 (2003)
Cohen, W., Shapire, R., Singer, Y.: Learning to order things. J. Artif. Intell. Res. **10**, 243–270 (1999)
Cortes C., Vapnik, V. (1995) Support–vector networks. Mach. Learn. J. **20**, 273–297 (1995)
Domshlak, C., Joachims, T.: Unstructuring user preferences: Efficient non-parametric utility revelation. In: Conference on Uncertainty in Artificial Intelligence (UAI), Edinburgh, Scotland (2005)
Doyle, J. Prospects for preferences. Comput. Intell. 20(2):111–136 (2004)
Doyle, J., Thomason, R.H.: Background to qualitative decision theory. AI Mag. **20**(2), 55–68 (1999)
Doyle, J., Wellman, M.: Representing preferences as ceteris paribus comparatives. In: Proceedings of the AAAI Spring Symposium on Decision-Theoretic Planning, pp. 69–75. Stanford, CA (1994)
Fishburn, P.C. The Foundations of Expected Utility. Reidel, Dordrecht (1982)
Freund, Y., Iyer, R., Schapire, R.E., Singer, Y.: An efficient boosting algorithm for combining preferences. J. Mach. Learn. Res. **4**, 933–969 (2003)
Goldsmith, J., Lang, J., Truszczynski, M., Wilson, N.: The computational complexity of dominance and consistency in CP-nets. In: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, pp. 144–149. Edinburgh, Scotland (2005)
Green, P.E., Krieger, A.M., Wind, Y.: Thirty years of conjoint analysis: Reflections and prospects. Interfaces. **31**(3), 56–73 (2001)
Ha, V., Haddawy, P.: A hybrid approach to reasoning with partially elicited preference models. In: Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence, pp. 263–270. Morgan Kaufmann, Stockholm, Sweden (1999)
Haddawy, P., Ha, V., Restificar, A., Geisler, B., Miyamoto, J.: Preference elicitation via theory refinement. J. Mach. Learn. Res. **4**, 317–337 (2003)
Hansson, S.O.: Preference logic. In: Gabbay, D.M., Guenthner, F. (eds.) Handbook of Philosophical Logic, IInd edn, vol. 4, pp. 319–394. Kluwer, Dordrecht (2001a)
Hansson, S.O. The Structure of Values and Norms. Cambridge University Press, Cambridge (2001b)
Herbrich, R., Graepel, T., Obermayer, K.: Large margin rank boundaries for ordinal regression. In: Advances in Large Margin Classifiers, pp. 115–132. MIT Press, Cambridge, MA (2000)
Herfert, M., La Mura, P.: Estimation of consumer preferences via ordinal decision-theoretic entropy. Leipzig Graduate School of Management, Working Paper HHL-Arbeitspapier Nr. 64 (2004)
Joachims, T., Granka, L., Pang, B., Hembrooke, H., Gay, G.: Accurately interpreting clickthrough data as implicit feedback. In: ACM Conference on Research and Development in Information Retrieval (SIGIR), pp. 154–161. Salvador, Brazil (2005)
Keeney, R.L., Raiffa, H.: Decision with Multiple Objectives. Wiley, NY (1976)
Kimeldorf, G., Wahba, G.: Some results on tchebycheffian spline functions. J. Math. Anal. Appl. **33**, 82–95 (1971)
Krantz, D.H., Luce, R.D., Suppes, P., Tversky, A.: Foundations of Measurement. Academic, NY (1971)
La Mura, P.: Foundations of Multi-Agent Systems. Ph.D. thesis, Graduate School of Business, Stanford (1999)
La Mura, P.: Decision-theoretic entropy. In: Proceedings of the Ninth Conference on Theoretical Aspects of Rationality and Knowledge, pp. 35–44. Bloomington, IN (2003)
La Mura, P., Shoham, Y.: Expected utility networks. In: UAI-1999, pp. 367–373. Stockholm, Sweden (1999)
Lang, J.: Logical preference representation and combinatorial vote. Ann. Math. Artif. Intell. **42**(1–3), 37–71 (2004)
Linden, G., Hanks, S., Lesh, N.: Interactive assessment of user preference models: The automated travel assistant. In: Proceedings of the Sixth International Conference on User Modeling, pp. 67–78. Chia Laguna, Sardinia, Italy (1997)

McGeachie, M., Doyle, J.: Utility functions for ceteris paribus preferences. Comput. Intell. **20**(2), 158–217 (2004)
McJones, P.: Eachmovie collaborative filtering data set. DEC Systems Research Center (1997)
Packard, D.J.: A preference logic minimally complete for expected utility maximization. J. Philos. Logic. **4**(2), 223–235 (1975)
Pu, P., Faltings B.: Decision tradeoff using example critiquing and constraint programming. Constr. Int. J. **9**(4), 289–310 (2004)
Pu, P., Faltings, B., Torrens, M.: User-involved preference elicitation. In: Working notes of the Workshop on Configuration (IJCAI-2003). Acapulco, Mexico (2003)
Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA (1993)
Riedl, J., Konstan, J., Lam, S., Herlocker, J.: Movielens collaborative filtering data set. http://www.grouplens.org (2006)
Shearin, S., Lieberman, H.: Intelligent profiling by example. In: Proceedings of the Sixth International Conference on Intelligent User Interfaces. Santa Fe, NM, USA (2001)
Shoham, Y.: Conditional utility, utility independence, and utility networks. In: Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence, pp. 429–436. Morgan Kaufmann Publishers, San Francisco, CA (1997a)
Shoham, Y.: A symmetric view of probabilities and utilities. In: IJCAI-1997, pp. 1324–1329. Nagoya, Japan (1997b)
Vapnik, V.: Statistical Learning Theory. Wiley (1998)
von Wright, G.H.: The logic of preference reconsidered. Theory and Decisions **3**, 140–167. Reprinted in [von Wright, 1984] (1972)
von Wright, G.H.: Philosophical logic: Philosophical Papers, vol. 2. Cornell University Press, Ithaca, NY (1984)
Wahba, G.: Spline Models for Observational Data, vol. 59 of CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM (1990)
Wilson, N.: Extending CP-nets with stronger conditional preference statements. In: Proceedings of the Nineteenth National Conference on Artificial Intelligence. San Jose, CL (2004)

Authors' vitae

Carmel Domshlak is a Senior Lecturer at the Faculty of Industrial Engineering and Management in Technion. He received his Ph.D. in Computer Science from Ben-Gurion University in 2002 for his work on preference representation models. His research interests are in a synthesis of knowledge representation and computational techniques across the field of AI, and in particular, in modeling and reasoning about preferences, planning, and probabilistic reasoning.

Thorsten Joachims is an Assistant Professor in the Department of Computer Science at Cornell University. In 2001, he finished his dissertation on text classification with Support Vector Machines advised by Prof. K. Morik at the University of Dortmund. From there he also received his Diplom in 1997 with a thesis on WebWatcher, a browsing assistant for the Web. His research interests center on a synthesis of theory and system building in the field of machine learning, with a focus on machine learning for information access and natural language processing.