A Small Survey on Preference Languages and Preference Mining Techniques

Sandra de Amo
Faculdade de Computação - Universidade Federal de Uberlândia - Brazil- deamo@ufu.br

The research literature on preference modeling, reasoning and eliciting is extensive.
Different specification approaches have been proposed to model user preferences such as graphs, logical formalisms and SQL-like constructors. The approach of CP-Nets [2] uses a very simple graphical model which captures users qualitative conditional preference over tuples, under a ceteris paribus semantics. The approach of TCP-Nets [3] generalizes the CP-Nets by introducing the ability of expressing absolute and relative importance of object attributes in the graphical model. In [19] a simple logical framework is introduced for expressing conditional preference statements (cp-rules). It generalizes both the CP-Net and TCP-Net approaches, by relaxing the ceteris paribus semantics and allowing more expressiveness. This formalism is used in [7] in order to extend the SQL query language with preference support. In [5] a simple logical framework is proposed for expressing preferences by means of preference formulae. These formulae are incorporated into the Relational Algebra and into SQL through the operator winnow parameterized by a preference formula. This approach is very expressive and generalizes the CP-Nets, TCP-Nets and the cp-rules of [19].

From a more operational standpoint, [14] proposes to inductively construct complex preferences by means of various basic and intuitive preference constructors. These preference constructors are used to extend the SQL query language with preference support [15]. Unlike the previous approaches, the formalism of [14] does not allow to express contextual preferences. In the present paper, we follow the approach of [1] for expressing contextual preference rules. In this work, the formalism $i^+ \succ i^- | X$ for specifying contextual preference rules has been introduced and a method for incorporating the preference rules into the query-answering mechanism of a database system has been proposed. The preference rules are supposed to be available and the problem of how the rules are obtained is out of the scope of that paper. For a comprehensive survey on preference modeling and query languages with preference support see [18].

In what follows we focus on the literature related to the topic of preference elicitation.

Preference learning can be divided into two distinct problems: label ranking and object ranking. Label ranking aims at predicting an order over a set of labels for a given object [8, 11]. One can view the objects as tuples containing information about users: age, education, economic status, city, profession, etc. And the labels represents objects to rank. The problem of label ranking consists in discovering rules relating user’s personal information to the way they rank
labels. The work of [11] discusses the differences underlying both problems and proposes a method for label ranking consisting in training a set of binary classifiers. In [8] a method for label ranking learning adapted from a classical method for association rules discovery is proposed. On the other hand, object ranking aims at predicting which is the preferred object between two given objects. The present paper focus on this latter problem. In what follows, we discuss some related work following this line of research.

Methods for Preference Learning can be categorized following different criteria such as Preference Specification (qualitative or quantitative), Preference Semantics (the pareto model, conditional preference model), Application Domain (document retrieval, query answering filtering, product rating and sentiment analysis, etc). The techniques presented in this section are inherently distinct. Nevertheless they have a common main goal: given a pair of objects, to predict which one is the most preferred.

In a qualitative approach, preferences are specified by a compact set of preference rules from which a preference relation can be inferred. The method we propose in this paper follows a qualitative approach. Some other qualitative approaches are [10, 12]. In [10] the authors propose a technique for mining user preferences whose underlying model is the pareto preference model. Such preference rules are obtained from log data generated by the server when the user is accessing a web site. Another approach to preference mining is presented in [12]. In this work the authors propose using preference samples provided by the user to infer an order on any pair of tuples in the database. Such samples are classified into two categories, the superior and inferior samples and contain information about some preferred tuples and some non-preferred ones. From these rules, an order is inferred on the tuples in the database. The underlying preference model is the pareto preference model as in [10]. In this model, preferences are not conditional or contextual, that is, preferences on values of attributes do not depend on the values of other attributes. In other words, our contextual preference model is more expressive.

Another intuitive preference model (but, non-contextual) is the lexicographic preference model (LPM). Unlike the pareto model, the LPM assumes an order of importance on the attributes that describe the objects in a domain and uses this order to make preference decisions. In [17], a greedy variable permutation algorithm has been proposed to produce one of the LPMs that is consistent with the user’s preference samples, if one exists. In [21], the authors propose to extract a collection of different lexicographic models from a set of preferences using a democratic voting system in order to approximate the target model.

Concerning the topic of mining conditional or contextual preference rules, [16] proposes a method for mining a CP-Net model from a set of preferences supplied by the user. Like in our approach, preference samples are represented by ordered pairs of objects. The goal is to identify a target preference ordering with a binary-valued CP-net by interacting with the user through a small number of queries. Our method is more flexible and general since it is able to compare sets of objects (i.e., transactions with different sizes) and can be also adapted to
compare tuples over a relational schema. Moreover, our method is flexible enough to deal with contradictory rules: [16] focuses on learning acyclic CP-Nets (with no contradictions).

In contrast with the above papers, where preferences are specified following a qualitative approach, in [4] and [6] algorithms for mining quantitative preferences are proposed. In these works preferences are specified by a score function and the main goal is to find automatically a prediction rule which assigns a score to each tuple of the database. The mining task in this approach is sometimes called learning to rank. Several efficient methods for learning to rank have been proposed so far in the information retrieval domain, including Rank SVM [13], RankBoost [9], RankNet [4] and AdaRank [20]. In all these methods, the learning task is formalized as classification of object pairs in two classes: correctly or incorrectly ranked. Different classification techniques are employed such as Support Vector Machine (Rank SVM), Boosting (AdaRank, RankBoost) and Neural Network trained by a Gradient Descent algorithm (RankNet).

References