Preference Modeling and Mining for Personalization

Seung-won Hwang
Pohang University of Science and Technology (POSTECH), Korea

INTRODUCTION

As near-infinite amount of data are becoming accessible on the Web, it becomes more important to support intelligent personalized retrieval mechanisms, to help users identify the results of a manageable size satisfying user-specific needs. Example case studies include major search engines, such as Google and Yahoo, recently released personalized search, which adapts the ranking to the user-specific search context. Similarly, e-commerce sites, such as Amazon, are providing personalized product recommendation based on the purchase history and user browsing behaviors. To achieve this goal, it is important to model user preference and mine user preferences from user behaviors (e.g., click history) for personalization. In this article, we discuss recent efforts to extend mining research to preference and identify goals for the future works.

BACKGROUND

Traditional modeling for user preference can be categorized into (1) quantitative and (2) qualitative approaches. In the quantitative approach, given a set of data objects D, a utility function F assigns a numerical score $F(o)$ for an object o in D. This utility score may aggregate scores on one (i.e., uni-attribute model) or more (i.e., multi-attribute model) *attributes* $F(a_1, \ldots, a_n)$, when $o = (a_1, \ldots, a_n)$. For instance, the utility of a house with *price = 100k* and *size = 100 square foot* can be quantified by a user-specific utility function, e.g., *F = size/price*, into a score, such that houses maximizing the scores, e.g., with largest size per unit price, can be retrieved.

Alternatively, in the qualitative approach, the preference on each object is stated in comparison to other objects. That is, given two objects x and y, instead of quantifying preferences into numerical scores, users simply state which one is more preferred, denoted as $x > y$ or $y > x$. Alternatively, users may state indifference $x \sim y$. Compared to quantitative modeling requiring users to quantify numerical scores of all objects, qualitative modeling is considered to be more intuitive to formulate (Payne, Bettman, & Johnson, 1993), while less efficient for evaluating the absolute utility of the specific object, as such evaluation requires relative comparisons to all other objects. Meanwhile, qualitative approach may also aggregate multiple orderings. Optimal results from such aggregation is formally defined as pareto-optimality as stated below.

Definition 1 (Pareto-optimality). A tuple x dominates another tuple y if and only if as $x > y$ or $x \sim y$ in all the given orderings.

MAIN FOCUS

Major components of enabling personalized retrieval can be identified as (1) preference modeling, (2) preference mining, and (3) personalization, each of which will be discussed in the following three subsections.

Preference Modeling

As discussed in the prior section, preferences are modeled typically as (1) quantitative utility function (Fishburn, 1970; Keeney & Raiffa, 1976) or (2) qualitative utility orderings (Payne et al., 1993). Personalization is guided by preferences represented in these two models, to retrieve ideal data results that maximize the utility. To maximize quantitative utility, *ranking query* (Guentzer, Balke, & Kiessling, 2000; Fagin, Lotem, & Naor, 2003) of returning few highly preferred results has been studied, while to maximize qualitative utility, *skyline query* (Börzsönyi, Kossmann, & Stocker, 2001; Godfrey, Shipley, & Gryz, 2007) of returning pareto-optimal objects not less preferred to (or "dominated" by) any other object based on the given

qualitative orderings, as we will discuss in detail in the personalization section.

Preference Mining

While user preference can be explicitly stated, in the form of a utility function or total orderings, such formulation can be too complicated for most end-users. Most applications thus adopt the approach of mining preferences from implicit feedbacks. In particular, preference mining from user click logs has been actively studied. Intuitively, items clicked by users can be considered as preferred items, over the items not clicked, which suggests *qualitative preference* of the specific user. More recently, the problem of using such qualitative reference information to infer *quantitative preference* utility function has been studied (Joachims, 2002; Radlinski & Joachims, 2005). These works adopt machine-learning approach to use qualitative orderings as training data to mine an underlying utility function.

Alternatively to *offline mining* of preferences from user logs, system may support dynamic and incremental preference elicitation procedures to collect additional user preference information and revise the result ranking. For *quantitative preference*, Yu, Hwang, and Chang (2005) studied adopting selective sampling to provide users with sample objects to provide feedbacks on, based on which the system collects information on user preferences and applies it in the retrieval process. To enable *online mining*, such sampling was designed to maximize the learning effectiveness such that the desired accuracy can be reached with the minimal user feedbacks. More recently, Joachims and Radlinski (2007) proposed to augment offline mining with online user elicitation procedure. For *qualitative preference*, Balke, Guentzer, and Lofi (2007) studied this online mining process to incrementally revise the skyline results, which was later extended to discuss a sophisticated user interface to assist users in the cooperative process of identifying partial orderings (Balke, Guentzer, & Lofi, 2007b).

Personalization

Once the preference is identified, we use it to retrieve the personalized results with respect to the preference.

For *quantitative preference*, the problem of efficient processing of ranking queries, which retrieve the results with the maximal utility score has been actively studied, pioneered by Algorithm FA (Fagin, 1996). Following works can be categorized into the two categories. First, more works followed to FA to be optimal in a stronger sense, by improving the stopping condition such that the upper bounds of the unseen objects can be more tightly computed, as presented in (Fagin et al., 2003). Second, another line of works follows to propose algorithms for various access scenarios, beyond FA assuming the sorted accesses over all predicates (Bruno, Gravano, & Marian, 2002; Hwang & Chang, 2005).

For *qualitative preference*, skyline queries are first studied as maximal vectors in (Kung, Luccio, & Preparata, 1975) and later adopted for data querying in (Börzsönyi et al., 2001) which proposes three basic skyline computation algorithms such as block nested loop (BNL), divide-and-conquer (D&C), and B-tree-based algorithms. Tan, Eng, and Ooi (2001) later study progressive skyline computation using auxiliary structures such as bitmap and sorted list. Kossmann, Ramsak, and Rost (2002) next propose nearest neighbor (NN) algorithm for efficiently pruning out dominated objects by iteratively partitioning the data space based on the nearest objects in the space. Meanwhile, Papadias, Tao, Fu, and Seeger (2003) develop branch and bound skyline (BBS) algorithm with I/O optimality property and Chomicki, Godfery, Gryz, and Liang (2003) develop sort-filter-skyline (SFS) algorithm leveraging pre-sorted lists for checking dominance condition efficiently. More recently, Godfrey, Shipley, and Gryz (2005) propose linear elimination-sort for skyline (LESS) algorithm with attractive average-case asymptotic time complexity, i.e., $O(d \cdot n)$ for d-dimensional data of size n.

More recently, there have been research efforts to combine the strength of these two query semantics. While skyline queries are highly intuitive to formulate, this intuitiveness comes with price of returning too many results especially when the dimensionality d of data is high, i.e., "curse of dimensionality" problem (Bentley, Kung, Schkolnick, & Thompson, 1978; Chaudhuri, Dalvi, & Kaushik, 2006; Godfrey, 2004) Recent efforts address this problem by narrowing down the skylines by ranking them and identifying the top-k results, which can be categorized into the following two groups of approaches: First, *user-oblivious ranking approach* leverages skyline frequency metric (Chan et al., 2006) which ranks each tuple in the decreasing order of the number of subspaces and in which the tuple is a skyline and k-dominances (Chan et al., 2006) which identifies k-dominant skylines as the common skyline