Progressive Ranking Based on a Dominance List

Yann Loyer
yann.loyer@prism.uvsq.fr
Isma Sadoun
issa@prism.uvsq.fr
Karine Zeitouni
karine.zeitoun@prism.uvsq.fr

PRiSM, CNRS UMR 8144
Université de Versailles Saint Quentin
45 avenue des État-Unis, 78035 Versailles, France

ABSTRACT
Preference queries aim at increasing personalized pertinence of a selection. The most famous ones are the skyline queries based on the concept of dominance introduced by Pareto. Many other dominances have been proposed. In particular, many weaker forms of dominance aim at reducing the size of the answer of the skyline query. In most cases, applying just one dominance is not satisfying as it is hard to conciliate high pertinence, i.e., a strong dominance, and reasonable size of the selection. We propose to allow the user to decide what dominances are reliable, and what priorities between those dominances should be respected. This can be done by defining a sequence, eventually transfinite, of dominances. According to that sequence, we propose operators that compute progressively the ranking of a dataset by successive applications of the dominances without introducing inconsistencies. The principle of progressive refinement provides a great flexibility to the user that can not only dynamically decide to stop the process whenever the results satisfies his/her wishes, but can also navigates in the different levels of ranking and be aware of the level of reliability of each successive refinement.

1. INTRODUCTION
Considerable attention has recently been paid to preference queries. Those queries aim to improve the pertinence of information retrieval that may be different from one user to another. They take into account user's preferences and have been studied following two different ways [4]. The first approach personalizes a given query by expanding it to include preferences. The second approach uses explicit preference operators in the query, such that the skyline operator [1] which is based on the concept of dominance or efficiency introduced by Pareto.

Considering a set of alternatives that can be compared with respect to a finite set of criteria, Pareto defined an alternative $A$ to be more efficient than another one $B$ (or to dominate $B$) if there is at least one criterion that suggests to prefer $A$ to $B$ while there exists none that suggests the contrary. The set of optimal alternatives, i.e. those that are not dominated by any other one, is called the frontier of Pareto. The skyline operator computes that frontier. In the context of high dimensional databases, skyline queries alone do not provide an efficient decision support. It is therefore necessary to refine the selection. Different approaches have been proposed to overcome that limitation. The main idea consists in introducing more comparability by defining other, mostly weaker, dominance relations. To name a few : $e$-dominance [5], K-dominance [2], dominance-back [6] and quasi-dominance [3]. The relevance of the different dominance relations is obviously disputable and depends on the context and/or the user. More generally, the dominance relation could even be defined by the user itself or be obtained from queries to experts, communities of users or web services. The dominance could even be obtained by the integration of information coming from different, eventually inconsistent, sources, using different combinations relying on operators such as set operators or other specific operators (e.g., see [4]). In fact, a dominance relation should simply be defined as a binary relation over the set of tuples. We propose to generalize that reasoning to any given set of binary relations over a same set of tuples. The user decides of (a) a selection, eventually transfinite, in the set of dominance relations of those he wants to use, (b) values for the parameters of each dominance relation that requires some, and (c) a strict total order, called preferences chain, over the set of relations he selected. Guided by that chain, we apply successively the dominance relations to refine progressively the answer set. Each step may allow new comparisons between tuples, but only between tuples that were considered incomparable and ranked at the same level by the precedent step.

In this paper, we define an operator that computes preferences chain guided rankings of a set of tuples (that can eventually be the skyline set). It applies successive dominance relations in a way that each application of one more dominance refines the ranking provided by the precedent one. The principle of progressive refinement provides a great flexibility to the user that can not only determine priorities between dominances he decides to rely on, but can also dynamically decide to stop the process whenever the results satisfies his wishes, or even navigates in the different levels of ranking, being aware of the level of reliability of each successive refinement. Finally, we provide experimental results that show the effectiveness and the efficiency of our algorithms.
2. RANKING REFINEMENT

Let \( R(d_1, \ldots, d_n) \) denote a database relation schema with \( n \) attributes where each attribute \( d_i \) takes values from a numerical domain \( \text{dom}(d_i) \). Let \( d = \{d_1, \ldots, d_n\} \) be the set of attributes of \( R \) and \( \text{dom}(d) \) be the domain of \( d \), defined by \( \text{dom}(d) = \text{dom}(d_1) \times \ldots \times \text{dom}(d_n) \). We use \( t \) to denote a tuple \((u_1, u_2, \ldots, u_n) \in \text{dom}(d)\) of \( R \), and \( r \) to denote a relation or dataset on \( R \), i.e. a set of tuples in \( \text{dom}(d) \). Let \( R^* \) be the set of datasets on \( R \).

Definition 1 (Dominance relation). A dominance relation over a dataset \( r \) is a binary relation over \( r \times r \).

Dominance relations are also called qualitative preference relations. The dominance relation that leads to the skyline partition is called traditional dominance denoted TD [1].

Definition 2 (Dominance relation). A dominance relation on \( R \) is a binary relation \( \preceq \) such that for any tuple \( t \in R \), \( \preceq \) is reflexive, transitive, and antisymmetric.

Definition 3 (Maximal element). A tuple \( t \) is maximal if for any tuple \( u \), if \( t \preceq u \), then \( t = u \).

Definition 4 (Maximal-based selection \( \max_{\theta} \)). Let \( \theta \) be a dominance relation over a dataset \( r \). The maximal-based selection \( \max_{\theta}(r) \) is defined as the set of maximal elements of \( r \) w.r.t. \( \theta \).

Definition 5 (\( \theta \)-decomposition \( \Gamma_{\theta} \)). Let \( (r_0, \ldots, r_m) \) be an ordered partition of a dataset \( r \). Let \( \theta \) be a dominance relation over \( r \). The \( \theta \)-decomposition of \( (r_0, \ldots, r_m) \) w.r.t. \( \theta \), denoted \( \Gamma_{\theta}(r_0, \ldots, r_m) \), is the ordered partition of \( r \) defined by \( \Gamma_{\theta}(r_0, \ldots, r_m) = (\Gamma_{\theta}(r_0), \ldots, \Gamma_{\theta}(r_m)) \).

Example 1. Figure 1 shows the example of \( \theta \)-decomposition. Each arrow color represents a dominance relation \( \theta_0 \) (black), \( \theta_2 \) (red), and \( \theta_3 \) (green). The decomposition of \( r \) w.r.t. \( \theta_1 \) (as illustrated by the blue shapes) is \( \gamma_0 = \max_{\theta_1}(r) = \{1, 10, 11\} \), \( \gamma_1 = \{2, 3, 4\} \), \( \gamma_2 = \{5, 13\} \), \( \gamma_3 = \{6, 7, 8, 9\} \), \( \gamma_4 = \{12\} \). So, \( \Gamma_{\theta_1} = \{(1, 10, 11), \{2, 3, 4\}, \{5, 13\}, \{6, 7, 8, 9\}, \{12\}\} \).

The idea is to refine progressively the ranking of the set of skyline tuples by successively applying the relations of a preferences chain. Thus, once the decomposition of \( r \) with respect to a dominance \( \theta_i \) has been computed, we propose to decompose the intermediate result using the next dominance \( \theta_{i+1} \). As the dataset \( r \) has already be pre-sorted based on \( \theta_i \), \( \theta_{i+1} \) should not be applied over the entire set \( r \) but only within the different subsets of \( r \) in order to refine the ranking.

![Figure 1: Preference chain ranking process](image-url)
Example 2. Let $\gamma = \Gamma_{\theta_i}(r)$ of example 1.

\[
\hat{\Gamma}_2(\gamma) = \langle \{10, 11\}, \{1\}, \{3, 4\}, \{2\}, \{5, 13\}, \{6\}, \{7, 9\}, \{8\}, \{12\} \rangle
\]

Definition 6 (Preferences chain guided ranking). Let $\Theta = \langle \theta_1, \ldots, \theta_n \rangle$ be a preferences chain over a dataset $r$. The preferences chain guided ranking of $r$ w.r.t. $\Theta$ is defined as the sequence $(\text{rank}^{\Theta}_0)$, where $\text{rank}^{\Theta}_0 = r$, $\text{rank}^{\Theta}_{n+1} = \hat{\Gamma}_{\theta_{n+1}}(\text{rank}^{\Theta}_n)$ for any successor ordinal $n$, and $\text{rank}^{\Theta}_\alpha = \max_\preceq \{\text{rank}^{\Theta}_n \mid n < \alpha\}$ for any limit ordinal $\alpha$.

Relying on Theorem 2, the above ranking is a well-defined concept.

Theorem 3. Let $\Theta$ be a preferences chain over a dataset $r$. The preferences chain guided ranking $(\text{rank}^{\Theta}_n)$ of $r$ w.r.t. $\Theta$ is a non-decreasing sequence w.r.t. $\preceq$ that reaches its limit, which is an ordered partition of $r$, in a finite number of steps.

We can now define a new operator that computes an ordered partition by progressive refinement.

Definition 7 (Preferences chain guided ranking). Let $\Theta$ be a preferences chain over dom($d$). The $\Theta$-ranking operator $\text{Rank}^{\Theta}_d$ associates to the relation $r$ in $R^d$ the limit of the preferences chain guided ranking $(\text{rank}^{\Theta}_n)$ of $r$ w.r.t. $\Theta$.

Example 3. Let $\Theta = \langle \theta_1, \theta_2, \theta_3 \rangle$.

$\text{Rank}^{\Theta}_d(r) = \Gamma_{\Theta_3}(\hat{\Gamma}_{\Theta_2}(\hat{\Gamma}_{\Theta_1}(r)))$

$= \langle \{10, 11\}, \{1\}, \{3, 4\}, \{2\}, \{5, 13\}, \{6\}, \{7, 9\}, \{8\}, \{12\} \rangle$

3. EXPERIMENTS

Experimental Settings. We evaluate the quality and the cost of $\text{Rank}^{\Theta}_d$ using the NBA data, which are NBA player statistics from 1946 to 2009 (http://databasesBasketball.com), with 21671 records over 17 attributes. In our test, we use two preferences chains: the one is based on the quasi-dominance denoted $QD$, while the other is based on the k-dominance denoted $KD$. Within each chain, we increase the indifference threshold $q$ for the $QD_n$, and decrease the dimension number $k$ for the $KD_n$. Progressive filtering and ranking capacity. Figure 2 represents the decomposition of $r$ after each step of the ranking guided by the preferences chain $QD_n$ of a set of 28 skyline tuples computed on a sample of 100 tuples in the NBA database. We can observe the progressive refinement of the ordered partitioning of the data set. Each line, from TD to QD($q^8$), shows the more refined partition of the set obtained after the application of one more dominance relation. The user can stop at any step or continue the ranking. In this figure, we chose to stop at QD($q^8$) only for a better illustration purpose. Figure 3 illustrate size of first subset of the ranking after each step w.r.t. $QD_n$.

Performance. Figure 4 represents the overcost of the refinement step by $\hat{\gamma}(r)$ for each successive dominance in $QD_n$. The overcost decreases since just smaller subsets are refined in each step. Figure 5 concerns the ranking guided by the preferences chain $KD_n$. The highest curve represents the cost of the total cumulated runtime of computation for each dominance in all the subsets of the ordered partition. Similarly, the lowest curve represents the cost of the total runtime for ranking all the subsets once the dominance is computed. As expected from Theorem 3, the overcost of each supplementary dominance converges to 0 as the ordered partition becomes progressively finer and the number of comparisons to be tested smaller. Note that the most expensive part of the computation is for the first dominances of the chain. While not necessary, it will be preferable for the user, from a performance point of view, to begin its chain with dominances such as Pareto dominance that satisfies some properties such as transitivity or anti-monotonicity that can be used to optimize their computation.

4. CONCLUSION

We propose a formal framework for progressive ranking of skylines sets with respect to users’ preferences. Our approach, is very flexible as it allows the user (a) to choose and order different relations of preference according to the reliability and priority he give to each of them, and (b) to decide to stop the progressive filtering as soon as the result satisfies him. We provide not only the formal framework but also experimental results which validate our approach.

5. REFERENCES