Progressive Ranking Based on a Dominance List

Yann Loyer
yann.loyer@prism.uvsq.fr

Isma Sadoun
issa@prism.uvsq.fr

Karine Zeitouni
karine.zeitouni@prism.uvsq.fr

PRiSM Labs, CNRS & Versailles university
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 : ϵ-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 $R(d_1, ..., d_n)$ denotes a database relation schema with n attributes where each attribute d_i takes values from a numerical domain $dom(d_i)$. Let $d = \{d_1, ..., d_n\}$ be the set of attributes of R and $dom(d)$ be the domain of d, defined by $dom(d) = dom(d_1) \times ..... \times dom(d_n)$. We use t to denote a tuple $(u_1, u_2, ..., u_n) \in dom(d)$ of R, and r to denote a relation or dataset on R, i.e. a set of tuples in $dom(d)$. 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 set is called traditional dominance denoted TD [1].
For a given dominance relation θ, one usually selects as "best" tuples with respect to θ those that are not dominated by any other tuple with respect to θ. But we consider a different definition of *maximality*. The usual definition asserts that a tuple is maximal if it is not dominated by any other one. If this is acceptable for pre-orders such as TD, it is not appropriate anymore for cyclic relations. For instance, suppose that a small subset s of r form a cycle w.r.t. a dominance relation and that no tuples in s is dominated by a tuple in $r \setminus s$ while any tuple in $r \setminus s$ is dominated by a tuple in s. In that case, there is no maximal tuple. We believe that cycles should be seen as set of equivalent tuples, i.e. elements that can not be preferred to each other (in our example, s should be the set of maximal elements). To this end, we will use the classical notion of transitive closure in order to derive a pre-order from each dominance relationships. The transitive closure is computed only for the tuples belonging to the cycle.

DEFINITION 2 (PARTIAL TRANSITIVE CLOSURE θ^+). *Let θ be a dominance relation over a dataset r. The partial transitive closure of θ over $s \subset r$, denoted θ_s^+, is the binary relation over s such that $\forall (t, t') \in s^2$, $\theta_s^+(t, t')$ iff $\exists (t_1, \ldots, t_v) \in s^v (t_1 = t \wedge t_v = t' \wedge \theta(t_1, t_2) \wedge \ldots \wedge \theta(t_{v-1}, t_v)))$.*

Relying on the transitive closure of a relation, we propose the definition of a new algebraic operator.

DEFINITION 3 (max_θ: MAXIMALITY-BASED SELECTION). *Let θ be a dominance relation over a dataset r. An element t in r is said to be maximal w.r.t. θ iff $\forall t' \in r$ $(\neg(\theta_r^+(t', t)) \vee \theta_r^+(t, t'))$. The maximality-based selection w.r.t. θ in r is the set of maximal elements of r w.r.t. θ is denoted $max_\theta(r)$.*

A tuple is maximal with respect to a dominance relation θ if and only if it dominates all the tuples that dominate it, i.e. iff $\forall t' \in r$ $(\theta_r^+(t', t) \Rightarrow \theta_r^+(t, t'))$. Of course, if it is not dominated by any tuple, then it is maximal. It is immediate that max_θ is a filter that selects some elements in r.

THEOREM 1. *Let θ be a dominance relation over a dataset r. $max_\theta(r) \subseteq r$ holds.*

First, we need to define the ranking with respect to a given dominance relation. That ranking is defined as an *ordered partition* of a dataset r, i.e. a *list* of *disjoint* subsets of r whose union is equal to r.

DEFINITION 4 (θ-DECOMPOSITION OPERATOR Γ_θ). *Let θ be a dominance relation over a dataset r. The decomposition of r w.r.t. θ, denoted $\Gamma_\theta(r)$, is defined as the ordered partition $\langle \gamma_0, \ldots, \gamma_p \rangle$ of r, where $\gamma_0 = max_\theta(r)$,*
$\gamma_i = max_\theta(r \setminus \bigcup_{j=0}^{i-1} \gamma_j)$ for $1 \leq i$, and $p = max\{i \mid \gamma_i \neq \emptyset\}$

That operator first computes the set γ_0 of maximal tuples with respect to θ. That set is the first set of the ordered partition. It represents the "first choice tuples" with respect to θ. Then it removes those selected tuples from the original set r. It computes the set γ_1 of maximal tuples in the remaining set r with respect to the partial transitive closure over that set. The resulting set is the second set of the partition and represents the "second choice tuples". The computation is iterated until there is no more tuples into the original set.

EXAMPLE 1. *Figure 1 shows the example of θ-decomposition. Each arrow color represents a dominance relation θ_i(θ1 is black, θ2 is red and θ3 is green). The decomposition of r w.r.t θ1 (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, $\Gamma_{\theta 1} = \langle \{1, 10, 11\}, \{2, 3, 4\}, \{5, 13\}, \{6, 7, 8, 9\}, \{12\} \rangle$.

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 θ_i has been computed, we propose to decompose the intermediate result using the next dominance θ_{i+1}. As the dataset r has already be pre-sorted based on θ_i, θ_{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.

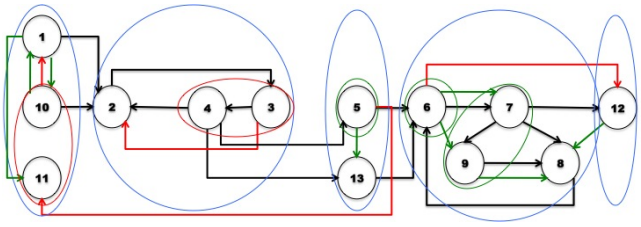

Figure 1: Preference chain ranking process

DEFINITION 5 (GENERALIZED θ-DECOMPOSITION $\hat{\Gamma}_\theta$). *Let $\langle r_0, \ldots, r_m \rangle$ be an ordered partition of a dataset r. Let θ be a dominance relation over r. The decomposition of $\langle r_0, \ldots, r_m \rangle$ w.r.t. θ, denoted $\hat{\Gamma}_\theta(r_0, \ldots, r_m)$, is the ordered partition of r defined by $\hat{\Gamma}_\theta(r_0, \ldots, r_m) = \langle \Gamma_\theta(r_0), \ldots, \Gamma_\theta(r_m) \rangle$.*

Let $\mathcal{O}(S)$ be the set of *lists of disjoint subsets* of a set S. Note that if $\langle x_1, \ldots, x_n \rangle$ in $\mathcal{O}(S)$ is such that $\bigcup_{1 \leq i \leq n} x_i = S$, then $\langle x_1, \ldots, x_n \rangle$ is an ordered partition of S. Let the order $\preceq$ over $\mathcal{O}(S) \times \mathcal{O}(S)$ be defined by $\langle x_1, \ldots, x_n \rangle \preceq \langle y_1, \ldots, y_m \rangle$, read $\langle y_1, \ldots, y_m \rangle$ is *finer* than $\langle x_1, \ldots, x_n \rangle$, iff for all y_i, y_j in $\langle y_1, \ldots, y_m \rangle$ such that $i \leq j$, there exist x_j, $x_{j'}$ in $\langle x_1, \ldots, x_n \rangle$ such that $y_i \subseteq x_j$ and $y_{i'} \subseteq x_{j'}$ and $i' \leq j'$. The following result asserts that the application of the operator $\hat{\Gamma}_\theta$ on an ordered partition is a refinement of that partition.

THEOREM 2. *Let $\langle r_0, \ldots, r_m \rangle$ be an ordered partition of a dataset r. Let θ be a dominance relation over r. Then $\langle r_0, \ldots, r_m \rangle \preceq \hat{\Gamma}_\theta(r_0, \ldots, r_m)$ holds.*

Finally, applying successively the decomposition operator will provide the user a global ranking as refined as possible

with respect to the sequence of dominance relations that he selected and ordered.

EXAMPLE 2. *Let* $\gamma = \Gamma_{\theta 1}(r)$ *of example1.*
$\hat{\Gamma}_{\theta 2}(\gamma) = \langle \Gamma_{\theta 2}(\{1,10,11\}), \Gamma_{\theta 2}(\{2,3,4\}), \Gamma_{\theta 2}(\{5,13\}), \Gamma_{\theta 2}(\{6,7,8,9\}), \Gamma_{\theta 2}(\{12\})\rangle$.
$\hat{\Gamma}_{\theta 2}(\gamma) = \langle \{10,11\}, \{1\}, \{3,4\}, \{2\}, \{5,13\}, \{6,7,8,9\}, \{12\}\rangle$

DEFINITION 6 (PREFERENCES CHAIN GUIDED RANKING). *Let* $\Theta = \langle \theta_1, \ldots, \theta_l \rangle$ *be a preferences chain over a dataset* r. *The* preferences chain guided ranking *of* r *w.r.t.* Θ *is defined as the sequence* $\langle rank_n^{\Theta} \rangle$, *where* $rank_0^{\Theta} = r$, $rank_{n+1}^{\Theta} = \hat{\Gamma}_{\theta_{n+1}}(rank_n^{\Theta})$ *for any successor ordinal* n, *and* $rank_{\alpha}^{\Theta} = max_{\preceq}\{rank_n^{\Theta}, n < \alpha\}$ *for any limit ordinal* α.

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

THEOREM 3. *Let* Θ *be a preferences chain over a dataset* r. *The preferences chain guided ranking* $\langle rank_n^{\Theta} \rangle$ *of* r *w.r.t.* Θ *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* Θ *be a preferences chain over* $dom(d)$. *The* Θ-ranking operator $Rank_{\Theta}$ *associates to the relation* r *in* R^* *the limit of the preferences chain guided ranking* $\langle rank_n^{\Theta} \rangle$ *of* r *w.r.t.* Θ.

EXAMPLE 3. *Let* $\Theta = \langle \theta 1, \theta 2, \theta 3 \rangle$.
$Rank_{\Theta}(r) = \hat{\Gamma}_{\theta 3}(\hat{\Gamma}_{\theta 2}(\Gamma_{\theta 1}(r)))$
$= \langle \{10,11\}, \{1\}, \{3,4\}, \{2\}, \{5\}, \{13\}, \{6\}, \{7,9\}, \{8\}, \{12\}\rangle$

3. EXPERIMENTS

Experimental Settings. We evaluate the quality and the cost of $Rank_{\Theta}$ 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 $\hat{\Gamma}_{\theta}(r_i)$ 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

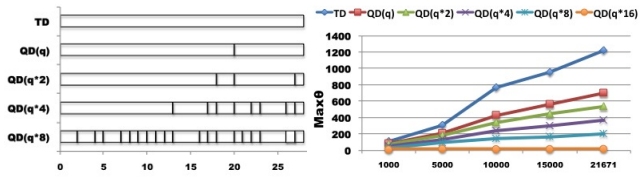

Figure 2: Refinement **Figure 3: Max Vs |DB|**

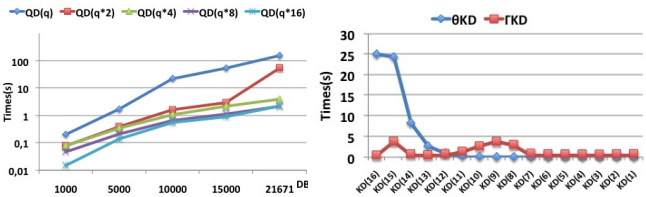

Figure 4: Perf of Max(θ) **Figure 5: Ranking Vs θ**

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

[1] Stephan Börzsönyi, Donald Kossmann, and Konrad Stocker. The skyline operator. ICDE, 2001.

[2] Chee Yong Chan, H. V. Jagadish, Kian-Lee Tan, Anthony K. H. Tung, and Zhenjie Zhang. Finding k-dominant skylines in high dimensional space. SIGMOD, 2006.

[3] J. Figueira, V. Mousseau, and B. Roy. Electre methods. Springer Verlag, 2005.

[4] Kostas Stefanidis, Georgia Koutrika, and Evaggelia Pitoura. A survey on representation, composition and application of preferences in database systems. ACM TODS, 36(3):19, 2011.

[5] Tian Xia, Donghui Zhang, and Yufei Tao. On skylining with flexible dominance relation. ICDE, 2008.

[6] Jing Yang, Gabriel Pui Fung, Wei Lu, Xiaofang Zhou, Hong Chen, and Xiaoyong Du. Finding superior skyline points for multidimensional recommendation applications. WWW, 2012.