Eliciting Matters – Controlling Skyline Sizes by Incremental Integration of User Preferences

Wolf-Tilo Balke[1], Ulrich Güntzer[2], Christoph Lofi[1]

[1] L3S Research Center, Leibniz University Hannover
Appelstr 4, 30167 Hannover, Germany
{balke, lofi}@l3s.de

[2] Institute of Computer Science, University of Tübingen
Sand 13, 72076 Tübingen, Germany
ulrich.güntzer@informatik.uni-tübingen.de

Abstract. Today, result sets of skyline queries are unmanageable due to their exponential growth with the number of query predicates. In this paper we discuss the incremental re-computation of skylines based on additional information elicited from the user. Extending the traditional case of totally ordered domains, we consider preferences in their most general form as strict partial orders of attribute values. After getting an initial skyline set our basic approach aims at interactively increasing the system's information about the user's wishes explicitly including indifferences. The additional knowledge then is incorporated into the preference information and constantly reduces skyline sizes. In fact, our approach even allows users to specify trade-offs between different query predicates, thus effectively decreasing the query dimensionality. We give theoretical proof for the soundness and consistence of the extended preference information and an extensive experimental evaluation of the efficiency of our approach. On average, skyline sizes can be considerably decreased in each elicitation step.

Keywords: skyline queries, partial order preferences, personalization.

1 Introduction

The problem that users cannot sensibly specify weightings or optimization functions for utility assessment of retrieval results has been considered for quite some time in the area of top-k queries and cooperative retrieval. Recently, the novel paradigm of skyline queries [6, 16, 15] has been proposed as a possible (if somewhat incomplete) answer. Skyline queries offer *user-centered* querying as the user just has to specify the basic predicates to be queried and in return retrieves the Pareto-optimal result set. In this set *all possible* best objects (where 'best' refers to being optimal with respect to any monotonic optimization function) are returned. Hence, a user cannot miss any important answer. However, this advantage of intuitive query formulation comes at a price: on one hand skylines are rather expensive to compute, on the other hand skylines are known to grow exponentially in size with an increasing number of predicate values [5].

In fact, experiments in [3] show that with as little as 5-6 independent query predicates usually already about 50 % of all database objects have to be returned as the skyline; clearly a prohibitive characteristic for practical uses. The problem even becomes harder if instead of totally ordered domains, partial order preferences on categorical domains are considered. In database retrieval, preferences are usually understood as partial orders [9, 13, 1] of domain values that allow for incomparability between attributes. This incomparability is reflected in the respective skyline sizes that are generally much bigger than in the totally ordered case. On the other hand such attribute-based domains like colors, book titles, or document formats play an important role in practical applications, e.g., digital libraries or e-commerce applications. As a general rule of thumb it can be stated that the more preference information (including its transitive implications) is given by the user with respect to each predicate, the smaller the average skyline set can be expected to be.

Building on our work in [2] in this paper we will discuss the incremental change of skyline sizes based on the newly elicited user preferences. Seeing preferences in their most general form as partial orders between domain values, this explicitly includes the case of totally ordered domains. After getting an (usually too big) initial skyline set our basic approach aims at interactively increasing the system's information about the user's wishes. The additional knowledge then is incorporated into the preference information and helps to reduce skyline sets. Our contribution thus is threefold:

- Users are enabled to specify *additional preference information* (in the sense of domination), as well as *equivalences* (in the sense of indifference) between attributes leading to an incremental reduction of the skyline. Here our system will efficiently support the user by automatically taking care that newly specified preferences and equivalences will never violate the consistency of the previously stated preferences (i.e. users will not encounter conflicts).
- Our skyline evaluation algorithm will allow specifying such additional information *within a certain predicate*. That means that more preference information about a predicate is elicited from the user. Thus the respective preference will be more complete and skylines will usually become smaller. This can reduce skylines to the (on average considerably smaller) sizes of total order skyline sizes by canceling out incomparability between attribute values.
- In addition, our evaluation algorithm will also allow specifying additional relations between preferences on *different predicates*. This feature allows defining the qualitative importance or equivalence of attributes in different domains and thus forms a good tool to compare the respective utility or desirability of certain attribute values. The user can thus express trade-offs or compromises he/she is willing to take and also can adjust imbalances between fine-grained and coarse preference specifications.

Especially the last contribution is of utmost importance and has not been considered in skyline query processing so far. It is the only way – short of dropping entire query predicates – to reduce the dimensionality of the skyline computation and thus severely reduce skyline sizes. Nevertheless the user stays in full control of the information specified and all information is only added in a qualitative way, and not by unintuitive weightings. We will prove in our experiments that using elicited preference information does indeed lead to the expected positive effect on skyline sizes.

2 A Skyline Query Use-Case and Related Work

To introduce the basic concepts of incremental preference enhancement, first we will present a short use case that will serve as a running example throughout the paper.

2.1 Basic Concepts of Partial Order Skyline Processing

Example: Consider a user deciding to buy a car. Usually he/she has preferences on at least some typical attributes like the car type, the color, the price, etc. Figure 1 shows three such preferences in the form of *strict partial orders*. These preferences can either be explicitly provided by the user together with the query or – what is more often the case – are provided as part of a user profile e.g., from typical usage patterns or previous user interactions. Sometimes they are also application/domain inherent like for example the preference on a lowest possible price for articles with the same characteristics in other respects. The skyline is then computed as the *Pareto-optimal set* over these preferences, e.g. a cheap red roadster dominates all expensive red, yellow or green car types, but for instance does not dominate any black car.

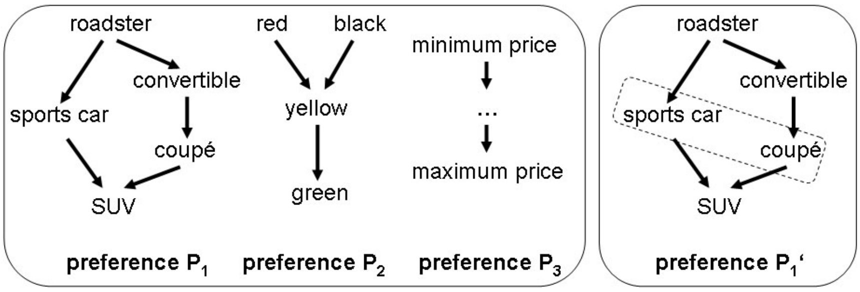

Figure 1. Three typical user preferences (left) and an enhanced preference (right)

Unlike for example the price preference that adheres to a naturally induced total order, preferences on categorical attributes will usually form only partial orders, expressing a user's indifference or indecisiveness. However, especially these attributes will increase skyline sizes, since the attribute's incomparability demands that they may all be part of the skyline. In fact tests in [2] show that partial order skylines sizes on a set of attribute values are on average about two orders of magnitude bigger than skylines where some total order has been declared on the same set of attributes. For instance a skyline over the preferences in figure 1 would contain all best red cars, as well as the best black cars. If a result size is too large to be manageable by the user, more specific information is required and has to be elicited.

Example (cont.): To reduce skyline sizes indifference can be reduced within each query predicate. A user can explicitly decide to *add a preference* to the current preference relation of a query predicate. For instance a user might state that he/she would rather have a black car than a red car and thus preference P_2 in figure 1 would be transformed into a total order by incrementing the object relationships in the already known preference relation by the relationships stating that black cars are generally preferred over all red cars with in all other respects equal or worse attribute values.

On the other hand a user might rather want to *state equivalences between attribute values*. Considering preference P_1 the user might express the equivalence between the sports car and the coupé like shown in preference P_1'. Implicitly this equivalence means that both car types are equally desirable and this also has consequences for the induced preference relation. For instance, the preference for convertible car types over coupés now also should imply a preference of convertible car types over sports cars. Stating the equivalence thus allows the user to express that sports cars and coupés are understood as indifferent choices, whereas the choice for a car type with removable top (such as a roadster or convertible) takes precedence for this user.

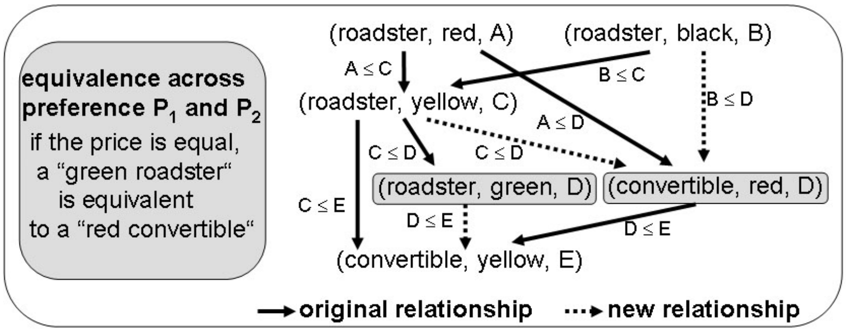

Figure 2. Original and induced domination relationships on object level

However, a user may not only have a feeling for relationships within a predicate, but also a feeling for the trade-offs he/she is willing to consider. The Pareto order describes the order of all possible 'packages' of predicate, i.e. induces an order between value tuples that are represented by at least one database object. Equivalences can be stated with respect to individual pairs of preferences, thus amalgamating preferences and effectively reducing the dimensionality of the skyline query.

Example (cont.): Declaring equivalences between different preferences is especially useful for stating differences in the amount of relaxation between preferences. Figure 2 shows the basic concept for our example, where every database object is a 3-tuple of car type, color and price. For example a user might find a relaxation of his/her color preferences less severe than a relaxation of the respective car's type. Consider for instance the roasters and convertibles. He/she could consider a green roadster (worst color) as equivalent to a red convertible (best color). The right hand side of Figure 2 shows the old and new domination relationships of different roasters to some convertibles. Note that after introducing the new equivalence all roadsters are considered better than convertibles (given that also the price is better or at least the same). We see the original domination relationships as defined by the Pareto order (black arrows) and those that were newly induced by the stated equivalence (dotted arrows). For example (given a better price) a previously incomparable black roadster now can be considered better than a red convertible, because it is better than a yellow roadster, which in turn is better than a green roadster that is considered equivalent to the red convertible.

Please note that such equivalences do not always have to lead to a 'lexicographical' ordering between preferences, but can also express more fine-grained relations between individual preferences, e.g. a certain amount in price is deemed to make up for one relaxation step in color. In any case, by eliciting new preferences or equivalences, the skyline size can never be increased. If any of the preference relations is enhanced by consistently adding more preference information, more domination relationships are possible in the Pareto order that is used for skyline evaluation. Hence the skyline size is bound to decrease monotonically.

2.2 Related Work

The problematic practical applicability of the skyline query paradigm in the face of exponentially growing result set sizes has been identified soon after its conception [1, 11]. To deal with this serious shortcoming several approaches have proposed the exploration of skylines in the form of user interaction. Since deriving a representative sample was proven NP-hard [14], this is done either by precomputing a 'skycube' that allows for OLAP-style interaction, or by exploiting user feedback on skyline samples to restrict the space of possible optimization functions. The latter approach [4, 3] aims at calculating a representative, yet manageable sample of the skyline to derive suitable utility functions for the user. Using these utility functions a top-k based approach can be performed that retrieves a manageable set of best objects, however, restricted to objects similar to those in the sample. In contrast, the skycube (or skyline cube) approach [20, 17] precomputes the skyline sets for various combinations of predicates such that a user can explore the skyline on-line e.g. by adding, dropping or aggregating predicates and consider the changes in the skyline. The major problem of this approach is the vast amount of expensive precomputations, which have to be repeated in the face of update operation to the data, see e.g. [19] for a discussion.

In human-computer interaction and AI, the importance of preference elicitation for a cooperative system behavior has already since long been recognized. Current approaches can be divided into those focusing on structural assumptions and those using feedback of users [21]. The first group features methods like value function elicitation [12] or the analytic hierarchy process [18]. Generally speaking they aim at composing utility value functions to rank a set of alternative choices. Assuming additive independence of predicates each individual predicate's utility is handled and then composed into a multi-dimensional utility function. A more flexible approach are CP-Networks [7] where the additive independence is replaced by conditional preferential independence allowing to use a set of totally ordered preference relations depending on the objects predicate values. Moreover, statistical approaches for eliciting preferences have been considered [8] where the elicitation process is modeled using a Markov-decision process over possible utility functions. In comparison, the approach in this paper is more general as it does not compose utility functions, but uses partial-ordered preferences that might even include several individual predicates.

Closest to our approach is the work in [10] examining theoretical properties of general incremental elicitation of partial and total order preferences. However, the work only examines possible preference collisions when combining (incrementally) or revising preferences in query modification and query evaluation.

3 Formalization of the Incremental Skyline Computation

To facilitate the incremental computation of skylines we need to formalize the preference and equivalence information that is exploited to calculate the respective skylines. Given a set of database objects O the preference relation stating the basic set of domination relationships (or for short: preferences) between individual database objects will be denoted as $P \subseteq O^2$. We will assume P to be free of cycles (i.e. consistent) and to induce a partial order between database objects: $\forall\, x, y \in O$ the expression $(x, y) \in P$ (or alternatively $x <_P y$) will denote that object y dominates object x with respect to all query predicates in the sense of Pareto optimality.

Similarly we will define $Q \subseteq O^2$ as an equivalence relation, i.e. a set of equivalences between database objects such that

a) Q is an equivalence relation (especially: is symmetric)
b) $Q \cap P = \emptyset$ (i.e. no equivalence in Q contradicts any strict preference in P)
c) $P \circ Q = Q \circ P = P$ (i.e. the domination relationships expressed transitively using P and Q should always already be contained in P)

We will call conditions a) to c) the *compatibility of equivalence relation Q with preference relation P* and use this characteristic to avoid inconsistencies between P and Q. Whereas it obviously does not make sense to specify equivalences that do not define an equivalence relation or directly contradict previously specified preference information, condition c) will have to be actively upheld by our incremental skyline computation algorithm. The idea of c) is that we start with only exact value equalities as equivalences (thus c) is trivially true) and then change Q and P accordingly for any equivalences that have been additionally specified.

Definition 1: (Expanded Preference and Equivalence Set)
Let O be a set of database objects, $P \subseteq O^2$ be a strict preference relation, $P^{conv} \subseteq O^2$ be the set of converse preferences with respect to P, and $Q \subseteq O^2$ be an equivalence relation that is compatible with P. Let further $S \subseteq O^2$ be a set of object pairs (called incremental preferences) such that

$$\forall\, x, y \in O: (x,y) \in S \Rightarrow (y,x) \notin S \text{ and } S \cap (P \cup P^{conv} \cup Q) = \emptyset$$

and let $E \subseteq O^2$ be a set of object pairs (called incremental equivalences) such that

$$\forall\, (x,y) \in O: (x,y) \in E \Rightarrow (y,x) \in E \text{ and } E \cap (P \cup P^{conv} \cup Q \cup S) = \emptyset.$$

Then we will define T as the transitive closure $T := (P \cup Q \cup S \cup E)^+$ and the expanded preference relation P^* and the expanded equivalence relation Q^* as

$$P^* := \{ (x,y) \in T \mid (y,x) \notin T\} \quad \text{and}$$
$$Q^* := \{ (x,y) \in T \mid (y,x) \in T\}$$

The basic intuition is that S and E contain the new preferences and equivalences that have been elicited from the user additionally to those given in P and Q. The only conditions on S and E are that they can neither directly contradict each other, nor are they allowed to contradict already known information. The sets P^* and Q^* then are the new preference/equivalence sets that incorporate all the information from S and E and that will be used to calculate the reduced skyline set. Definition 1 indeed results in the desired incremental skyline set as we will prove in theorem 1:

Theorem 1: (Correct Incremental Skyline Evaluation with P^* and Q^*)
Let P^* and Q^* be defined like in definition 1. Then the following statements hold:

1) P^* defines a strict partial order (specifically: P^* does not contain cycles)
2) Q^* is a compatible equivalence relation with preference relation P^*
3) $Q \cup E \subseteq Q^*$
4) The following statements are equivalent
   a) $P \cup S \subseteq P^*$
   b) $P^* \cap (P \cup S)^{conv} = \varnothing$ and $Q^* \cap (P \cup S)^{conv} = \varnothing$
   c) No cycle in $(P \cup Q \cup S \cup E)$ contains an element from $(P \cup S)$

   and from either one of these statements follows: $Q^* = (Q \cup E)^+$

Proof:
Let us first show two short lemmas:

Lemma 1: $T \circ P^* \subseteq P^*$
Proof: Due to T's transitivity $T \circ P^* \subseteq T \circ T \subseteq T$ holds. If there would exist objects $x, y, z \in O$ with $(x, y) \in T$, $(y, z) \in P^*$, but $(x, z) \notin P^*$, then follows $(x, z) \in Q^*$ because T is transitive and the disjoint union of P^* and Q^*. Due to Q^*'s symmetry we also get $(z, x) \in Q^*$ and thus $(z, y) = (z, x) \circ (x, y) \in T \circ T \subseteq T$. Hence we have (y, z), $(z, y) \in T \Rightarrow (y, z) \in Q^*$ in contradiction to $(y, z) \in P^*$. ■

Lemma 2: $P^* \circ T \subseteq P^*$
Proof: analogous to lemma 1 ■

ad 1) From lemma 1 directly follows $P^* \circ P^* \subseteq P^*$ and thus P^* is transitive. Since by definition 1 P^* is also anti-symmetric and irreflexive, P^* defines a strict partial order. ■

ad 2) We have to show the three conditions for compatibility:
a) Q^* is an equivalence relation. This can be shown as follows: Q^* is symmetric by definition, is transitive because T is transitive, and is reflexive because $Q \subseteq T$ and trivially all pairs $(q,q) \in Q$.
b) $Q^* \cap P^* = \varnothing$ is true by definition 1
c) From lemma 1 we get $Q^* \circ P^* \subseteq P^*$ and due to Q^* being reflexive also $P^* \subseteq Q^* \circ P^*$. Thus $P^* = Q^* \circ P^*$. Analogously we get $P^* \circ Q^* = P^*$ from lemma 2.
Since a), b) and c) hold, equivalence relation Q^* is compatible to P^*. ■

ad 3) Since $Q \subseteq T$ and Q is symmetric, $Q \subseteq Q^*$. Analogously $E \subseteq T$ and E is symmetric, $E \subseteq Q^*$. Thus, $Q \cup E \subseteq Q^*$. ■

ad 4) We have to show three implications for the equivalence of a), b) and c):
a) $\Rightarrow$ c): Assume there would exist a cycle $(x_0, x_1) \circ \ldots \circ (x_{n-1}, x_n)$ with $x_0 = x_n$ and edges from $(P \cup Q \cup S \cup E)$ where at least one edge is from $P \cup S$, further assume without loss of generality $(x_0, x_1) \in P \cup S$. We know $(x_2, x_n) \in T$ and $(x_1, x_0) \in T$, therefore $(x_0, x_1) \in Q^*$ and $(x_0, x_1) \notin P^*$. Thus, the statement $P \cup S \subseteq P^*$ cannot hold in contradiction to a).

c) $\Rightarrow$ b): We have to show $T \cap (P \cup S)^{conv} = \emptyset$. Assume there would exist $(x_0, x_1) \circ \ldots \circ (x_{n-1}, x_n) \in (P \cup S)^{conv}$ with $(x_{i-1}, x_i) \in (P \cup Q \cup S \cup E)$ for $1 \leq i \leq n$. Because of $(x_0, x_n) \in (P \cup S)^{conv}$ follows $(x_n, x_0) \in P \cup S$ and thus $(x_0, x_1) \circ \ldots \circ (x_{n-1}, x_n)$ would have been a cycle in $(P \cup Q \cup S \cup E)$ with at least one edge from P or S, which is a contradiction to c).

b) $\Rightarrow$ a): If the statement $P \cup S \subseteq P^*$ would not hold, there would be x and y with $(x, y) \in P \cup S$, but $(x, y) \notin P^*$. Since $(x, y) \in T$, it would follow $(x, y) \in Q^*$. But then also $(y, x) \in Q^* \cap (P \cup S)^{conv}$ would hold, which is a contradiction to b).

This completes the equivalence of the three conditions now we have to show that from any of we can deduce $Q^* = (Q \cup E)^+$. Let us assume condition c) holds.

First we show $Q^* \subseteq (Q \cup E)^+$. Let $(x, y) \in Q^*$, then also $(y, x) \in Q^*$. Thus we have two representations $(x, y) = (x_0, x_1) \circ \ldots \circ (x_{n-1}, x_n)$ and $(y, x) = (y_0, y_1) \circ \ldots \circ (y_{m-1}, y_m)$, where all edge are in $(P \cup Q \cup S \cup E)$ and $x_n = y = y_0$ and $x_0 = x = y_m$. If both representations are concatenated, a cycle is formed with edges from $(P \cup Q \cup S \cup E)$. Using condition c) we know that none of these edges can be in $P \cup S$. Thus, $(x, y) \in (Q \cup E)^+$.

The inclusion $Q^* \supseteq (Q \cup E)^+$ holds trivially due to $(Q \cup E)^+ \subseteq T$ and $(Q \cup E)^+$ is symmetric, since both Q and E are symmetric. ■

The evaluation of skylines thus comes down to calculating P^* and Q^* as given by definition 1 after we have checked their consistency as described in theorem 1, i.e. verified that no inconsistent information has been added. It is a nice advantage of our system that at any point we can incrementally check the applicability and then accept or reject a statement elicited from the user or a different source like e.g. profile information. Therefore, skyline computation and preference elicitation are interleaved in a transparent process.

For the actual skyline computation we rely on the algorithm given in [2] for customized Pareto aggregation. The customized Pareto operator already uses both preference and equivalence information for each predicate. The preference and equivalence information in our case is given by P^* and Q^* respectively.

Only for the predicates spanning across predicates we have to slightly adapt the customized Pareto aggregation. In case of a preference/equivalence connecting two preferences P_i and P_j and/or their respective equivalence sets Q_i and Q_j, we first amalgamate the two individual predicates by customized Pareto aggregation to a new preference P and equivalence set Q as follows (in the notation of [2]): $P :=$ Pareto(O, P_i, P_j, Q_i, Q_j) and $Q := \{((x_1, x_2),(y_1, y_2)) \mid (x_1, y_1) \in Q_i \text{ and } (x_2, y_2) \in Q_j\}$

After we have amalgamated the preferences as shown, we can easily insert all preference and equivalence information spanning both predicates and run the normal customized Pareto aggregation for skyline evaluation, however, with the advantage of reduced dimensionality. Moreover, since the new skylines will only get smaller we can restrict all incremental skyline computations to the already retrieved set in the previous step. Thus, the skyline is only once computed expensively over the full database and all subsequent steps are then only calculated over increasingly smaller data sets.

4 Experimental Section

In this section we evaluate the effects and implications of our approach. For a fair comparison several synthetic datasets and preference relations are generated randomly for each measurement series and the averages are reported. Throughout the tests random preferences mimicking realistic preference graphs are successively extended by pieces of equivalence information (thus introducing some new preference relations in P* and its transitive closure). We evaluate multiple scenarios with changing parameters to study general characteristics of our approach. The base parameters of each scenario, unless stated differently can be found in Table 1. (cf. experiments in [2]):

Table 1. Base parameters for the evaluation scenarios

Parameter	Value
Database Size	100,000
Distribution	uniform
Number of Query Predicates	6
Predicates' Domain Size (# distinct attribute values)	30
Preference Depth (longest path within graph)	8
Edge Degree (ration between graph nodes and edges)	1.2
Unconnected Degree (ratio between isolated and connected nodes in graph)	0.05

4.1 Influence of Incrementally Adding Equivalence Edges on the Result Size

In this scenario, we examine the average reduction of skyline size during the incremental addition of edges. Our claim is that adding more and more edges will decrease the size of the resulting skyline set significantly. This is especially true for adding equivalences between different predicates, i.e. amalgamating preferences. For performing this evaluation we considered uniform, normal and Zipf distributions of data. During the course of each run, up to ten valid edges (according to definition 1) are randomly inserted *into* or *between* preference relations (each case separately). After incrementally adding an edge, the resulting skyline size is measured. The resulting average sizes are shown for uniform distribution in Figure 3: the average skyline size was reduced in only ten steps to 73 % of its original size in the case of adding only edges within preference graphs and to 34 % using edges between different preferences. Our experiments for data sets following a Gaussian and Zipf distribution provide similar results and thus confirm them. In the Zipf case (at a skew of 0.7), however the initial skyline was already considerably smaller (about 49.000 compared to 62.000 objects) and hence also the decrease in skyline sizes were less pronounced.

4.2 Examination of the Normalized Result Size Reductions

To quantify the respective decline in skyline sizes we examined the behavior after adding each edge. Obviously, different edges can have a vastly different influence on the size reductions. There are some edges (e.g. between leave nodes) that will not contribute much, whereas other edges (e.g. connecting disjoint parts of a graph near the root) will be highly beneficial. Hence, this effect has to be studied under some suitable normalization. An obvious normalization that can be easily calculated is the number of edges that an incrementally added edge in a base preference actually causes to be inserted in the transitive closure of P^* and Q^* (which in turn form the base for the new skyline calculation). We thus calculated the decrease in skyline size as percentage of a single edge in the transitive closure. The observed mean value of 0.16 shows that per edge in the transitive close the skyline can be expected to decrease by about 0.16%. With a measured standard deviation of 0.13, however, this value is no adequate tool for predicting skyline reductions and a more sophisticated approach, involving more complex statistical characteristics of the data set (cf. e.g. [11]), will be necessary for accurately predicting result skyline reductions.

Therefore, we also checked the impact of new preference information for diverse preferences over the same set of data (i.e. how the actual shape of the preference varies the impact of new information). We measured the average absolute benefit of a single random additional equivalence and considered its distribution. The left hand side of Figure 4 shows our results. The impact of new information shows a mean of about 11247 objects and a standard deviation of 5473. Although it seems to resemble a Gaussian distribution, a Shapiro-Wilk test with a confidence of 95% fails.

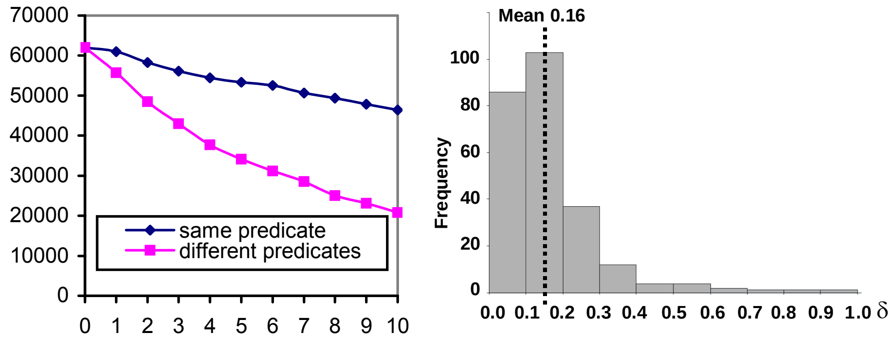

Figure 3. Result set sizes for incrementally added edges (left) and the frequency histogram of the distribution of the observed normalized size reductions δ (right).

4.3 Influence of Preference Depth on Skyline Result Set Sizes

Finally, we varied the number of additional edges over different preference depths (i.e. approaching a total order). The right hand side of figure 4 reports our results. Plotted are the average respective skyline sizes for 0, 3, 6 and 9 incrementally added edges between preference graphs over a dataset of 100,000 objects. With increasing preference depth, the result set sizes also decrease significantly due to the reduction of incomparable predicate values within the preferences. As the preferences more and more resemble a total order (which is reached at a depth of 30) the initial skyline becomes increasingly lean and the respective reductions by adding more information decrease. Adding more information thus is more important for 'bushy' preferences as opposed to total orders. In any case, also this experiment confirms that eliciting more information from the user leads to significantly diminished skyline sizes.

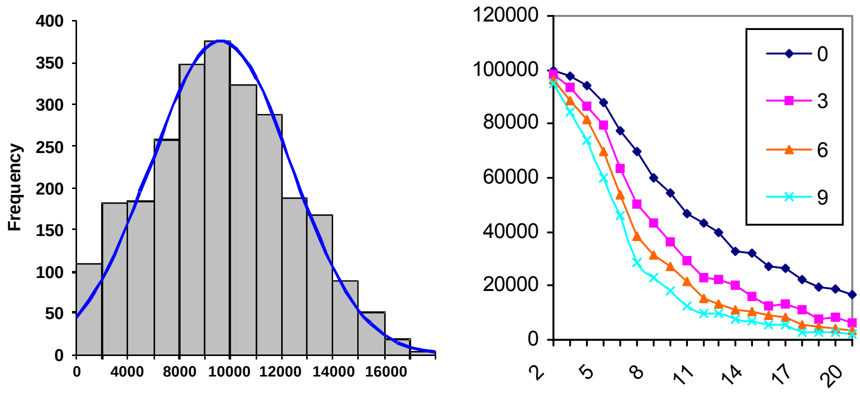

Figure 4. Absolute impact of new preference information (left) and skyline sizes for varying numbers of additional edges and preference depths.

5 Summary and Outlook

In this paper we have shown that unpractical skyline sizes can be controlled by eliciting more information from the user and incrementally recomputing the respective skylines. In our framework users are not only enabled to specify additional preference information (in the sense of domination), but also equivalences (in the sense of indifference) between attributes. Moreover, our skyline evaluation allows for specifying such additional information within a certain predicate and even between preferences on different predicates. In any case users are supported by automatically taking care that newly specified preferences and equivalences will never violate the consistency of any previously stated preferences and their implications. This feature allows defining the qualitative importance or equivalence of attributes in different domains and thus forms a good tool to compare the respective utility or desirability of attribute values: users can express compromises they are willing to take, and adjust imbalances between fine-grained and coarser preference specifications. Our experiments confirm that this indeed can reduce the skylines to the total order skyline sizes by canceling out incomparability and that usually only a few new relations are needed.

Our future work will especially focus on reducing the necessary recomputation steps for deriving the incremental skyline. Since all new information added is only of a local nature, also the new skyline can be expected only to change with respect to several attributes that were affected by the changes. This may lead to considerably reduced computation times for the incremental skyline.

Acknowledgments. Part of this work was supported by a grant of the German Research Foundation (DFG) within the Emmy Noether Program of Excellence.

6 References

1. W.-T. Balke, U. Güntzer. Multi-objective Query Processing for Database Systems. *Int. Conf. on Very Large Data Bases (VLDB),* Toronto, Canada, 2004.
2. W.-T. Balke, U. Güntzer, W. Siberski. Exploiting Indifference for Customization of Partial Order Skylines. *Int. Database Engineering and Applications Symp. (IDEAS)*, Delhi, India, 2006.
3. W.-T. Balke, J. Zheng, U. Güntzer. Efficient Distributed Skylining for Web Information Systems. *Int. Conf. on Extending Database Technology (EDBT)*, Heraklion, Greece, 2004.
4. W.-T. Balke, J. Zheng, U. Güntzer. Approaching the Efficient Frontier: Cooperative Database Retrieval Using High-Dimensional Skylines. *Int. Conf. on Database Systems for Advanced Applications (DASFAA),* Beijing, China, 2005.
5. J. Bentley, H. Kung, M. Schkolnick, C. Thompson. On the Average Number of Maxima in a Set of Vectors and Applications. *Journal of the ACM (JACM)*, vol. 25(4) ACM, 1978.
6. S. Börzsönyi, D. Kossmann, K. Stocker. The Skyline Operator. *Int. Conf. on Data Engineering (ICDE)*, Heidelberg, Germany, 2001.
7. C. Boutilier, R. Brafman, C. Geib, D. Poole. A Constraint-Based Approach to Preference Elicitation and Decision Making. *AAAI Spring Symposium on Qualitative Decision Theory*, Stanford, USA, 1997.
8. C. Boutilier. A POMDP Formulation of Preference Elicitation Problems. *National Conference on Artificial Intelligence (AAAI),* Edmonton, USA, 2002
9. J. Chomicki. Preference Formulas in Relational Queries. *ACM Transactions on Database Systems (TODS)*, Vol. 28(4), 2003.
10. J. Chomicki. Iterative Modification and Incremental Evaluation of Preference Queries. *Int. Symp. on Found. of Inf. and Knowledge Systems (FoIKS)*, Budapest, Hungary, 2006.
11. P. Godfrey. Skyline Cardinality for Relational Processing. *Int Symp. on Foundations of Information and Knowledge Systems (FoIKS)*, Wilhelminenburg Castle, Austria, 2004.
12. R. Keeney, H. Raiffa. Decisions with Multiple Objectives: Preferences and value tradeoffs. *Cambridge University Press,* 1976.
13. W. Kießling. Foundations of Preferences in Database Systems. *Int. Conf. on Very Large Databases (VLDB)*, Hong Kong, China, 2002.
14. V. Koltun, C. Papadimitriou. Approximately Dominating Representatives. *Int. Conf. on Database Theory (ICDT)*, Edinburgh, UK, 2005.
15. D. Kossmann, F. Ramsak, S. Rost. Shooting Stars in the Sky: An Online Algorithm for Skyline Queries. *Int. Conf. on Very Large Data Bases (VLDB)*, Hong Kong, China, 2002.
16. D. Papadias, Y. Tao, G. Fu, B. Seeger. An Optimal and Progressive Algorithm for Skyline Queries. *Int. Conf. on Management of Data (SIGMOD),* San Diego, USA, 2003.
17. J. Pei, W. Jin, M. Ester, Y. Tao. Catching the Best Views of Skyline: A Semantic Approach Based on Decisive Subspaces. *Int. Conf. on Very Large Databases (VLDB)*, Trondheim, Norway, 2005.
18. T. Satty. A Scaling Method for Priorities in Hierarchical Structures. *Journal of Mathematical Psychology*, 1977.
19. T. Xia, D. Zhang. Refreshing the sky: the compressed skycube with efficient support for frequent updates. *Int. Conf. on Management of Data (SIGMOD),* Chicago, USA, 2006.
20. Y. Yuan, X. Lin, Q. Liu, W. Wang, J. Yu, Q. Zhang. Efficient Computation of the Skyline Cube. *Int. Conf. on Very Large Databases (VLDB)*, Trondheim, Norway, 2005.
21. L. Chen, P. Pu. Survey of Preference Elicitation Methods. *EPFL Technical Report IC/2004/67*, Lausanne, Swiss, 2004.