Eliciting Customer Wishes using Example-Based Heuristics in E-Commerce Applications

Christoph Lofi, Wolf-Tilo Balke
Institut für Informationssysteme
University of Braunschweig
Mühlenpfordtstraße 23
Braunschweig, Germany
lofi@ifis.cs.tu-bs.de, balke@ifis.cs.tu-bs.de

Ulrich Güntzer
Institute of Computer Science
University of Tübingen
Sand 13
Tübingen, Germany
ulrich.guentzer@informatik.uni-tuebingen.de

***Abstract*—The ubiquitous access to information via the Web has changed our daily life and business practices. E-commerce applications allow for a wide variety of vendors to compete in a world-wide market. But with a growing number of vendors, also the amount of available offers is increasing, which in turn leads to an information flood that may severely hamper the user experience. Skyline algorithms as introduced by the information systems community, promise to winnow suboptimal offers from electronic marketplaces. For the amount of Web data and the typical interaction style, however, the result sets are still too large and hard to manage. Recently, first approaches to integrate human decision processes like compromises or trade-offs have been designed. In this paper we will build on these approaches and introduce a novel heuristic into the skylining paradigm that not only allows for convenient Web-style user interaction, but also focuses searches on semantic clusters of offers. Thus, the view on interesting clusters is refined, whereas less interesting clusters are strongly reduced in size and strictly focused on only the outstanding items.**

Keywords - personalization, e-commerce, preference elicitation, skyline queries, trade-offs.

1. INTRODUCTION

Today's possibility to access data on a world-wide scale has changed the way of commercial interactions. E-commerce portals and electronic marketplaces offer platforms for vendors and buyers from different areas and countries. But the flood of different offered products also leads to severe problems with respect to the manageability and comparability of individual items on offer. The traditional way of dealing with this variety is generally provided by comparison shopping portals which tap into different Web sources and aggregate an overview of all offers. In the technical sense these portals mostly rely on the paradigm of quantitative compensation of attributes using ranking approaches such as Top-K retrieval (for a comprehensive survey see e.g. [1]). Here either the system or the user individually weighs and ranks the attributes, thus providing a utility function summarizing benefit of each individual item.

Unfortunately, this compensation feature is of a purely *quantitative* nature and thus not very intuitive to use. For instance, assume a user wants to buy a new laptop computer. Using an internet shopping portal, he/she is usually faced with several thousand offered notebooks. Now, for choosing between laptops the semantics of a possible utility function like "*0.4 * price + 0.8 * RAM_size + 0.2 * performance*" cannot be grasped naturally by most users, because determining the correct weightings to reach a desired ranking in the sense of personalization is far from trivial; for an example see Fig. 1.

Recently, the skyline paradigm (see e.g., [2,3]) has been successfully established in the information systems community as a *qualitative* technique for personalized filtering of suboptimal items from large datasets. In order to facilitate this task, skylining has been designed following the well-known concept of *Pareto* optimality as known from economic theory. Pareto optimality is able to derive qualitative *domination relationships* from a set of basic attribute orders (i.e. user preferences) between database items. A database item can be said to dominate another item, if it features better or equal attribute values with respect to all user provided attribute preferences. Hence, Pareto dominated objects always can safely be excluded from any query's result set. Let us come back to our e-shopping example: consider two almost identical machines, only differing with respect to price and processor. Assume that one model is slightly more expensive and furthermore features a slower processor. In this case, there is no sensible reason for preferring this on all accounts inferior choice over the cheaper and more powerful laptop, and thus should according to the skyline semantics be removed from the result set to ease the user's cognitive load, as well as foster trust in the system's recommendations.

However, the skyline paradigm does not come entirely without drawbacks: while performance issues have been mostly solved by efficient indexing [4,5] or multi-processor extensions [6,7], weaknesses with respect to semantic characteristics and the manageability of skyline sets remain. For

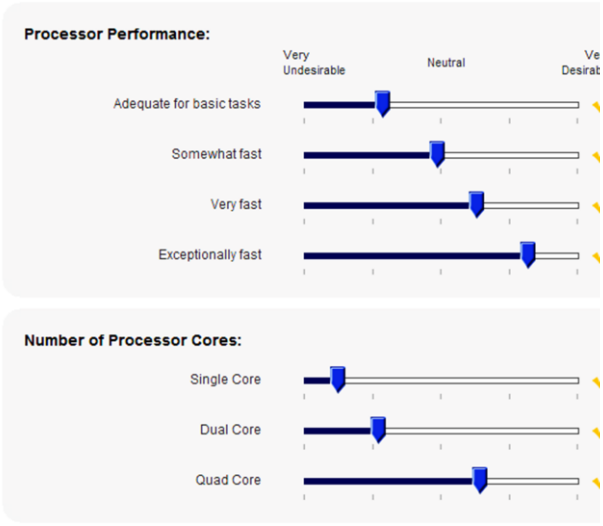

Fig. 1. Quantitative weighting with Sliders (from http://www.myproductadvisor.com)

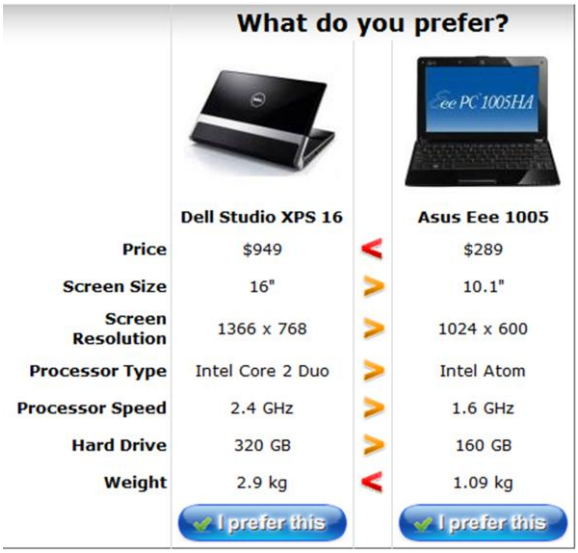

Fig. 2. Qualitative object comparisons

example, it is often reported that the actual size of skyline sets is still far from being manually manageable by users: being swamped by up to 30% of the original database items for just 5 to 10 attributes (see [8-10]) is no rarity. This effect is a result of the fact that often preferences are conflicting on correlated attributes (e.g., a user is looking for a notebook with "*low* price, but *high* quality" or "*low* weight, but *large* screen"), thus rendering items incomparable with respect to the basic Pareto semantics. In particular, the ability of *compensating* (in the sense of individual utility) between attributes is missing when using skyline queries.

In order to bridge the gap between quantitative Top-K queries and qualitative skyline approaches, recently a comprehensive theory on qualitative trade-offs for extending the semantics of skylines[11,12] was proposed. To explain the basic concepts, reconsider the previous sample user trying to buy a laptop computer: after performing a skyline query with all 'natural' preferences for the notebook domain (e.g. cheap price, fast processor, large display, low weight, etc.), he/she may still be presented with many Pareto-optimal notebooks. This is because the result set spans over the full range of available notebook types: powerful desktop replacements, overall good office notebooks, subnotebooks with high mobility, down to extremely low cost netbooks. Obviously, all these choices are part of the skyline result set as they all excel in at least one desirable attribute, or offer a good compromise. Actually, this problem is a common phenomenon: different product classes within a given domain almost always feature this intrinsic incomparability – if a given product class was on all counts inferior to some other, the respective class can be expected to fail on the market.

Unfortunately, skyline queries in the basic form do not allow for further personalization, i.e. focusing the skyline by removing or penalizing product categories which are undesired by an individual user. Qualitative *trade-offs* [11], on the other hand allow compensating between different (product) features and bias the skyline result towards each user's individual preferences by introducing new domination relationships. In particular, a trade-off relating two example items may compensate between different classes: is a user considering a netbook willing to pay for the superior performance of a subnotebook, but also to accept its generally higher weight and price? In any case, the newly introduced relationships will always respect the original Pareto semantics: users still do not need to exactly quantify the compensation function or weightings, but just state simple attribute-based preferences on (two or more) real world items. Still, having users provide trade-offs just using object comparisons usually leads to over-specified relationships that do not have a significant effect on the skyline size.

Furthermore, the trade-off skyline approach relies on users providing trade-offs explicitly, and rarely any attention was paid to the trade-off elicitation process. While a statistical trade-off suggestion scheme was presented in [13], other options for obtaining user trade-offs remain unexplored.

On the other hand, so called 'example critiquing interfaces' are commonly explored for eliciting user preferences in e-commerce scenarios. First results do not only show that their usage leads to the desired items, but it also leads to a positive user experience [14].

In this paper, we will therefore introduce an innovative heuristic approach for eliciting qualitative trade-offs relying on simple example critiques while still ensuring a smooth user experience and promoting an effective personalization of the skyline result set. In a nutshell, the user is faced with

simple decisions between skyline objects (see Fig. 2), while the system heuristically deduces trade-offs from these decisions. Moreover, we will show that using this elicitation process, exceptionally good deals in each category always remain part of the skyline and thus the user's choices are not overly restricted. The benefit of our novel technique is manifold:

- It allows to design state of the art e-commerce applications and portals that by using skylining filter only Pareto optimal choices to enhance the usage experience
- It allows for rich personalization facilities that can easily be integrated into applications and portals, because the induced ranking builds on the same mathematical foundation as the original skylining framework
- The resulting user interaction is of a simple nature and can easily displayed on nowadays mobile devices
- Finally, it enables the design and use of dynamic interfaces to work interactively with the system for focusing result sets with respect to the users' personal needs and thus combats the information flood in today's global markets

The remainder of the paper is structured as follows: in section 2 we showcase the elicitation heuristic using an example scenario and in section 3, the necessary theoretical foundations and notations for trade-off skylines are provided. Finally, the heuristic is formalized, justified, and discussed in section 4. We conclude this paper with evaluations in chapter 5, and final discussions in chapter 6.

2. Eliciting User Trade-Offs Using Heuristics

A natural approach for eliciting customer preferences and also individual preference trade-offs is to rely on object comparisons: in contrast to explicitly asking for attribute weights and utility functions for ranking, users are simply presented with some real world example objects from the database and decide which one they prefer (e.g., see Fig. 2), i.e. which offer is the more appealing deal. In fact, advanced e-commerce portals like MSN's product-centered shopping portals already allow comparing technical specifications and/or photos (for the example of cars see http://www.autos.msn.com/research/compare). Similar inter faces are also incorporated in the Ciao shopping portal of Bing's search engine (http://www.ciao.de/). However, these interfaces come without the possibility to integrate the user's elicited preferences into the subsequent search process – they provide only means for manual comparisons by users themselves. This example clearly shows that comparison interfaces are already in place, but the technology and algorithms needed to individually influence the user experience based on such comparisons is not yet provided.

In fact, while the approach of evaluating object comparisons to induce a personalized ranking is easy for users from a cognitive point of view, it is far from trivial to reliably deduce usable information from a set of object comparison decisions. This has brought forward the area of preference learning: in particular, such comparisons have been used to "learn" user preference for quite some while [15]. Usually, the idea is to employ machine learning algorithms (in particular support vector machines) to approximate the weights of the user's top-k style utility function [16]. Unfortunately, it has been shown that quite a large number of object comparisons is necessary to estimate good utility functions, thus those approaches usually either stay on a theoretical level, or are restricted to domains in which users cater for the required number of comparisons by heavy interaction (e.g. web search result ranking, [17]).

In this paper, we will showcase how objects comparisons can be used to intuitively elicit user trade-offs and how these trade-offs can then be heuristically exploited to gain additional, if somewhat more general trade-offs to further reduce result set sizes. These trade-offs can be used to further focus a personalized skyline result set into a so-called trade-off skyline. Furthermore, we will also justify this heuristic by showing that the resulting semantic is analogous to some corresponding utility function. That means in particular that our qualitative trade-offs are indeed an alternative to quantitative approaches with respect to the order of result objects, albeit with improved ease of use.

A. *Trade-Off Skylines*

Trade-Off skylines [11] are an extension of the popular Pareto skyline paradigm, which allows for qualitative compensation between attribute values. Pareto skylines rely on the concept of Pareto optimality for filtering suboptimal database items. Any query result item can be safely ignored, if there exists some item that simply shows better or equal values with respect to *all* query attributes: we say the ignored item is '*dominated'*. This is indeed a very intuitive concept: if for example two car dealers in the neighborhood offer exactly the same model (with same warranties, etc.) at different prices, why should one want to consider the more expensive car? However, the skyline concept is not without drawbacks: it offers no facilities for further compensating between attribute dimensions, and thus the personalization options are rather limited and the result sets can grow unmanageably large [18].

Trade-Off skylines remedy these problems by providing an additional layer of personalization, and thus reduce the size and increase usefulness of the skyline result set.

B. *Offering Trade-Off Choices*

In contrast to [11] which relied on providing trade-offs explicitly, in this work trade-off should be extracted heuristically from object comparisons. In order to facilitate this process, it has first to be decided which objects should be chosen for user comparisons. Different approaches for selecting candidates from the skyline are possible:

a) Candidates are selected randomly or the user freely chooses any two tuples him-/herself. However, this imposes an unnecessary cognitive load, while on average also yielding low effects with respect to skyline reduction.

b) Candidates are selected considering the correlation of attribute domains (as e.g., suggested [13]). This approach selects example objects based on the degree of attribute correlation, assuming that feedback on objects with opposing attribute values within anti-correlated domains will result in a most informative trade-off. This approach is easy to implement as it does only rely on the attribute values.

c) Candidates are selected employing semantic clustering (as e.g., described in [19]). Here, data objects are first semantically clustered by their perceived similarity. The resulting clusters usually represent different product groups, e.g. netbooks or desktop replacements. Selecting the prominent representatives of clusters as candidates for comparison thus leads to strong and semantically meaningful trade-offs. However, for modeling "perceived similarity", a vast effort in text mining on document collections extracted from internet sources like e.g. technology blogs, professional review sites, or online magazines is required. By employing natural language processing techniques and extensive statistics, "similarity" from the user's point of view is approximated.

In the following, we will assume that the trade-offs are directly provided by users.

C. Trade-Off Specification

As an example, assume in Fig. 2 that a user prefers an Asus EEE (the EEE is a netbook) over a Dell Studio XPS (an all-round notebook). Taking all the provided information into account directly results in the canonical trade-off "($289, 10.1'', 1024×600, Atom, 1.6 GHz, 160 GB, 1.09kg) is better than ($949, 16'', 1366×768, Core 2D, 2.4 GHz, 320 GB, 2.9kg)". This trade-off, however, may be quite weak due to over-specification: usually a trade-off will allow domination between any object equal or better than its preferred candidate and any other object equal or worse than the second candidate (additionally to the domination relationships already induced by the Pareto-optimality criterion of the skyline computation). In the worst case, there are no such objects besides the EEE and Studio XPS notebook themselves, and thus only one object, the Studio XPS, would be removed from the skyline. Also, this trade-off does not capture why the user preferred one notebook over the other – and this is exactly the missing semantics our heuristics-based approach aims to provide.

D. Heuristically Extending Trade-Off Information

Consider why a user might prefer a netbook over an all-round notebook: the prominent advantage of a netbook is its cheap price and high mobility (light weight, small form factor). In the case at hand, these two aspects are strong enough (at least for this user) to compensate all performance disadvantages the netbook faces in all other respects. Furthermore, this means that, because the user is willing to accept all the resulting disadvantages, he/she will definitely also accept fewer disadvantages given that the same advantages are offered. This heuristic idea naturally leads to multiple conceptual trade-offs offering views on the user's implicit concepts driving the current trade-off selection. In the case of our previous trade-off for example: "($289, 10.1'', 1.09kg) is better than ($949, 16'', 2.9kg)" (cheap & mobile compensates small screens) or "($289, Atom, 1.6GHz, 1.09kg) is better than ($949, Core 2D, 2.4 GHz, 2.9kg)" (cheap & mobile compensates slow and inferior CPUs).

These considerations are to be formalized in the following section.

3. Theoretical Foundations

To be self-contained, in this section we will briefly summarize the necessary theoretical foundations of Pareto trade-offs and introduce the notions required to formalize our heuristic. For a comprehensive discussion on skyline trade-offs, please refer to [11] which provides the theoretical foundations for modeling, validating and computing arbitrary trade-off skylines. However, the paper does not provide any techniques for actual user interaction (e.g. user interface, trade-off elicitation, heuristics, etc.). This current paper extends the purely theoretical work in [11] with a general heuristic-based framework for intuitively obtaining trade-off information from users.

Assume a database relation $R \subseteq D_1 \times \ldots \times D_n$ on n attributes. Let A be the set of all available attributes $A := \{A_1, \ldots, A_n\}$. For each attribute A_i, a *base preference* P_i is provided as a *weak order* over D_i (please note that the limitation to weak orders is only chosen to simplify formal notations in this paper; of course all following theories can also be applied to more complex partial orders with equivalences). If some attribute value $a \in D_i$ is preferred over another value $b \in D_i$ with respect to P_i, this is written as $a >_i b$ (reads as "a dominates b wrt. to P_i"). Similarly, $a \approx_i b$ ("a is equal to b wrt. to P_i") and $a \gtrsim_i b$ ("a either dominates or is equal to b wrt. to P_i") can be defined. For example, in the preferences in Fig. 3, $"Core\ i7" >_3 "Turion\ X2"$ and $"Turion\ X2" \approx_3 "Core\ 2\ Duo"$ hold.

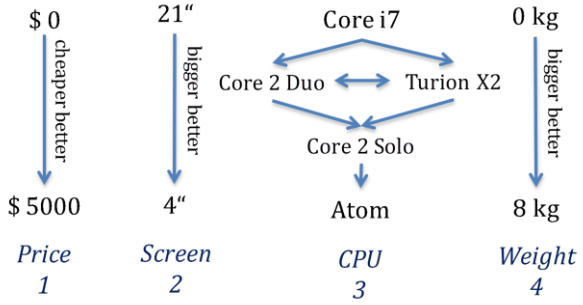

Fig. 3. Example Preferences for Notebooks as Partial Orders

Now, the skyline can be computed by removing all dominated objects from the result set. An object o_1 dominates an object o_2 (written as $o_1 \succ_P o_2$) if o_1 dominates or is equal to with respect to each attribute and dominates with respect to at least one attribute. This definition leads to the so called Pareto dominance:

Definition: Pareto dominance

$o_1 \succ_P o_2 \equiv \forall\, i \in \{1, \dots, n\}: o_{1,i} \succsim_i o_{2,i} \wedge$

$\exists\, i \in \{1, \dots, n\}: o_{1,i} \succ_i o_{2,i}$

Finally, trade-offs can be considered as a user preference between two partial tuples x and y, focusing usually only on a subset μ of all available attributes, i.e. $\mu \subseteq \{1, \dots, n\}$ and $x, y \in D_\mu$ with $D_\mu := \bigtimes_{i \in \mu} D_i$. Furthermore, the complement set $\bar{\mu} := \{1, \dots, n\} \setminus \mu$ is required for later considerations. Then, the trade-off t is denoted as $t := (x \rhd y)$. An example trade-off t_1 using the preferences of Fig. 3 could be $t_1 :=$(($290, 10.1'', 1.1kg) $\rhd$ ($940, 16'', 2.9kg)) with $\mu_1 := \{1, 2, 4\}$ and $\overline{\mu_1} := \{3\}$. In order to be valid, the two parts x and y of the trade-off need to be Pareto-incomparable, i.e. they don't dominate each other (if they did, then the trade-off is either superfluous as it would not provide any additional information over the base preference or it would directly contradict the base preferences).

Trade-offs further focus a skyline (i.e. remove unwanted objects) by introducing additional domination relationships. Simplified, a trade-off t will induce a trade-off domination relationship between any object o_1 and o_2 (denoted as $o_1 \succ_t o_2$) for which o_1 dominates or is equal to x with respect to μ, y is equal or dominates to o_2 with respect to μ, and finally o_1 dominates or is equal to o_2 with respect to $\bar{\mu}$. Refer to Fig. 4 for a visualization of this concept. Formally, a single trade-off domination is thus expressed as

$$o_1 \succ_t o_2 \equiv \begin{pmatrix} \forall i \in \mu: \left(o_{1,i} \succsim_i x_{1,i}\right) \wedge \left(y_{1,i} \succsim_i o_{2,i}\right) \\ \wedge\, \forall i \in \bar{\mu}: o_{1,i} \succsim_i o_{2,i} \end{pmatrix}$$

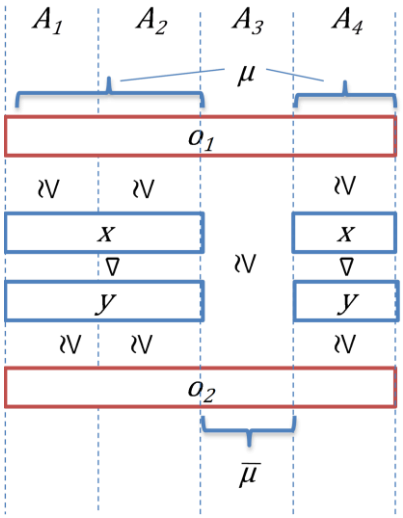

Fig. 4. Trade-Off Domination
Example of $o_1 \succ_t o_2$: Object o_1 dominates o_2 via the trade-off $(x \rhd y)$. The trade-off is defined on $\mu = \{1,2,4\}$; i.e. represents a multi-dimensional preference on the attributes A_1, A_2, and A_4. o_1 dominates or is equivalent to x with respect to those attributes and o_2 wrt. A_3. Likewise, y dominates or is equivalent to o_2 wrt. to A_1, A_2, and A_4.

As an example for a trade-off dominance, consider the previous trade-off t_1 =(($290, 10.1'', 1.1kg) $\rhd$ ($940, 16'', 2.9kg)).
This trade-off will result in the Asus EEE netbook (from Fig. 2) dominating, e.g. the Dell Studio which is thus removed from the result.

Unfortunately, computing skylines respecting multiple trade-offs is far from trivial as trade-off dominations may *chain*, e.g. o_1 dominates o_2 only when two trade-off t_1 and t_2 are present and evaluated consecutively. Complex interactions between multiple trade-offs as well as resulting evaluation algorithms are again covered in [11].

Finally, the conceptual trade-off heuristics introduced in section 2 can formally be expressed as follows:

Definition: conceptual trade-offs

For $t = (x \rhd y)$ on the index set μ, and $\mu_1 \subsetneq \mu$ with $\forall\, i \in \mu_1$: $x_i \preceq_i y_i$ then a conceptual trade-off t_1 can be defined as: $t_1 := (x|_{\mu_2} \rhd y|_{\mu_2})$ with $\mu_2 := \mu \setminus \mu_1$ (i.e. $x|_{\mu_2}$ is x restricted to all attributes not in μ_1).

This means: each trade-off $(x \rhd y)$ can be complemented by a set of conceptual trade-offs; these are generated by dropping any subset of attributes for which x shows inferior values to y. Semantically, the new trade-offs are more general versions of the user selection, i.e. trade-offs containing the same positive attributes (in which x dominates y) but fewer negative attributes (in which y dominates x).

4. Semantic Justification

In the following, we will argue that this heuristic is indeed semantically correct when assuming that a user does act according to a (generally unknown) *utility functions*. In particular, we show that domination relationships created by our conceptual trade-offs will not conflict with any utility function on the base preferences.

Preposition: Conceptual Trade-Offs do not contradict any monotonic utility function.

Proof: Let f be a monotonic *utility function* which represents a score for the aggregated usefulness of a database item from the user's point of view. Furthermore, we assume without loss of generality that all attribute values are already normalized with respect to some *scoring functions* (i.e. all $a_i \in [0, 1]$ with 1 being the best value and 0 being the worst, thus already encoding the user's base preferences usually given by weak orders). Also, all used functions are assumed to feature the correct arity. Then, we can safely conclude that whenever $t = (x \rhd y)$ on some attributes in

μ with $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_n)$ holds, then also $f(x) > f(y)$ holds.

Furthermore, let g and h be additional monotonic utility functions such that $f(x_1, \dots, x_n) = g\big(x_1, h(x_2, \dots, x_n)\big)$, i.e. higher-arity utility functions can be composed of lower-arity utility functions (semantically, this means that advantages/ disadvantages within one attribute can be numerically compensated by the aggregated advantages and disadvantages of all remaining attributes).
Thus, from $t = (x \rhd y)$ now also follows $\big(x_1, h(x_2, \dots, x_n)\big) > g\big(y_1, h(y_2, \dots, y_n)\big)$.

Now, we assume that $x_1 < y_1$ (i.e. $1 \in \mu_1$; and A_1 is one of the attributes which are dropped from the trade-offs as described in section 3). Then it can easily proved that due to the monotony of utility functions also $g\big(y_1, h(x_2, \dots, x_n)\big) > g\big(x_1, h(x_2, \dots, x_n)\big)$ follows (because $y_1 > x_1$). Combining this and the previous formula results to $g\big(y_1, h(x_2, \dots, x_n)\big) > g\big(y_1, h(y_2, \dots, y_n)\big)$. Again, by simply exploiting the monotony of utility functions, we obtain $h(x_2, \dots, x_n) > h(y_2, \dots, y_n)$. This fact, however, is equivalent to introducing a conceptual trade-off $(x|_{\mu\setminus\{1\}} \rhd y|_{\mu\setminus\{1\}})$. Furthermore, it is easy to see that any conceptual trade-off as described in section 3 can be constructed by permutating the attribute order and recursively repeating the previous steps. ■

This result shows that all conceptual trade-offs generated by the application of our heuristic on a given trade-off t are indeed compatible with any implicit or explicit utility function if the original trade-off t was also compatible with the function, i.e. conceptual trade-offs represent a subset of the semantics of utility functions in a purely qualitative manner. Combined with the fact that compared to constructing a user's utility function much less user-elicited information is necessary to construct trade-offs, conceptual trade-offs prove to be a powerful tool for personalization purposes.

5. Evaluation

To showcase the effect of trade-offs with respect to skyline size reduction we use a typical real world dataset. Our real world set is genuine crawled E-commerce web dataset and contains 998 notebook offers (http://www.shopping.com). This evaluation is performed similar to the evaluation in [13] in which an established a basic scenario for computing and evaluating trade-off skylines. However, the works in [13] are based on a significantly less general model for possible user trade-offs (e.g. only trade-offs on pairwise disjoint pairs of attributes are allowed). Besides providing formal definitions and algorithms for this limited trade-off model, it also contains some work on user interaction. However, the chosen approach differs significantly from the approach in our current work as it tries to suggest some trade-offs (according to the simplified underlying model) based on data statistics (e.g. degree of anti-correlation of attributes, data distribution, etc.). The user is presented with some (potentially unintuitive trade-offs) and may choose between them. Our current work has a somewhat similar base agenda (i.e. interacting with users to obtain a trade-off skyline; thus the following chapter is aligned to [13] to enable a comparison of approaches), but is based on completely different trade-off models (i.e. general and unrestricted trade-offs as introduced by [11]) and interaction paradigms; relying on simple object comparisons and deducing trade-offs instead of suggesting specific trade-offs based on statistics.

E. Trade-Off Suggestion

As arbitrarily chosen trade-offs will rarely have useful effects on the actual skyline size, we use sets of four trade-offs provided by a simple suggestion scheme: the base idea of most trade-off suggestion algorithms is to reduce incomparability induced by a high degree of anti-correlation between two attributes and their respective correlated attributes. Our notebook dataset features the following six numerical attributes: CPU speed, ram capacity, hard drive capacity, screen size, weight, and price. Based on the data, trade-offs can be straightforwardly suggested as follows: first, the two attributes with the strongest degree of anti-correlation are determined and their respective domain values are clustered. The attribute which shows clusters with a higher degree of separation and inner cohesion is selected as main attribute. Then the centroid values of the larger clusters form the base of one half of the suggested trade-offs, e.g. for notebooks the main attribute is screen size with 15", 14.1", 13.3", etc as clusters. These trade-offs' components are then expanded by the cluster's average values of all attributes which are correlated or anti-correlated with the main attribute, e.g. correlation coefficient $\rho \geq 0.5$ or $\rho \leq -0.5$. This approach is chosen to ensure compatibility to evaluations in [13] and provides trade-offs spanning four of the six attributes. Those trade-offs are thus "real" trade-offs which are already quite strong by themselves. Still, even those trade-offs can further benefit from application of the conceptual trade-off heuristic despite their limited size. Also note that the trade-offs used in this evaluation could also be directly generated by applying our heuristic to much weaker trade-offs resulting from a simple object comparison between representative notebooks of each group.

F. Setup and Results

Following trade-offs were considered: 1) a trade-off favoring a typical netbook over a typical subnotebook, 2) a trade-off favoring the typical subnotebook over a typical smaller office notebook (13.3" screen), 3) one trade-off favoring the smaller office notebook over a standard office notebook (15" screen), and finally 4) one trade-off favoring the standard office notebook over a typical desktop replacement machine (in short: the more mobile and cheaper, the better). Our results are as follows: the original skyline of the 998-tuple notebook dataset contained 201 items.

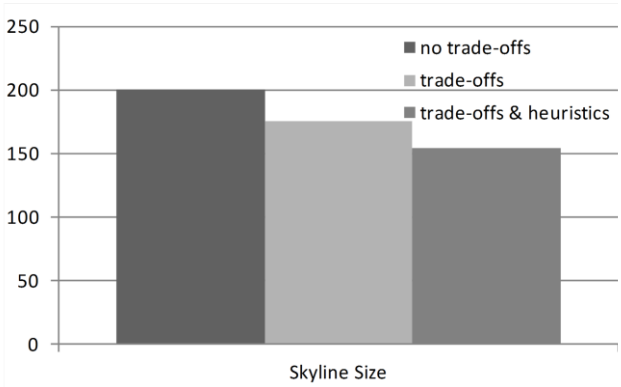

Fig. 5. Skyline Size (4 trade-offs)

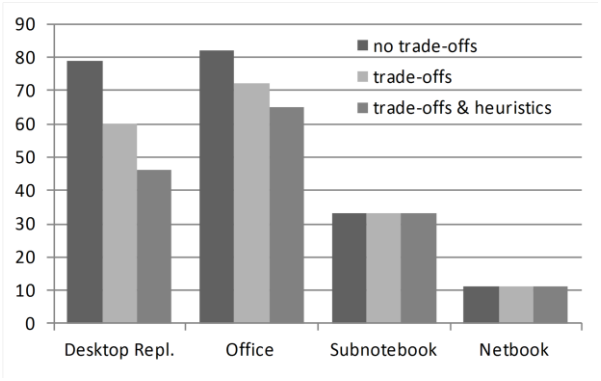

Fig. 6. Focusing of the Skyline

After adding the four initial trade-offs, the size decreases to 176 items. After applying the conceptual heuristics, the size further decreased down to 155 (see Fig. 5).

Furthermore, we examined which objects have been excluded from the skyline by incorporating the trade-offs. In order to perform this task, we classified each notebook in one of the following groups: desktop replacements, office notebooks, subnotebooks, and netbooks. The size of those groups is illustrated Fig. 6. It can be clearly observed that when adding our trade-offs, set of netbooks and subnotebooks remains stable due to their desirability. Only tuples from the desktop replacement and office notebooks group are excluded from the skyline (from 79 to 60 for desktop replacement, and from 82 to 72 for office notebooks). Please note that while these two groups are reduced in size, still the characteristics of Pareto optimality remain mostly intact (in contrast to just using a hard filter): exceptionally good offers in both desktop replacement and office notebook groups still remain in the result set.

By applying our heuristics, this trend continues: again, neither netbooks nor subnotebooks are removed; the number of desktop replacements is further reduced from 60 down to 46 items, and the number of office notebooks decreases from 72 to 65. Thus, the skyline does not only become more manageable by being reduced in size, but furthermore also focuses more strongly on the actual product groups which are of interest to the user.

Please note that some core characteristics of skyline queries still hold when incorporating trade-offs (or conceptional trade-offs) in strict skyline result sets: in contrast to simple and hard filters ("I definitely want a netbook and no desktop replacement"), the remaining desktop replacement notebooks are very good deals which still exhibit outstanding attribute combinations even when respecting the provided trade-offs. Thus, the general spirit of just removing sub-optimal choices and presenting the user with a wide selection of distinctively good database objects which is pursued by vanilla skylining is still maintained.

6. CONCLUSIONS

In this paper, we presented a strong heuristic framework which allows to deduce more information about user concepts during interaction in e-commerce applications. In particular, we relied on a qualitative preference framework extended by trade-off information and showed that simple comparisons of items can be extended by our heuristic to capture the semantics of a traditional ranking function. However, unlike eliciting individual weights for complex ranking functions, leading users through some well-chosen and simple item comparisons is a far more intuitive style of interaction on the Web. In particular, our presented heuristic is especially tailored for intuitive web-friendly user interactions: instead of providing complex trade-offs directly, users just need to provide basic feedback with respect to some simple object comparisons. Our approach thus minimizes the required cognitive overhead and allows for a smooth user experience.

Moreover, our framework leads to increased retrieval effectiveness and opens up the effective use of trade-off skylines for Web applications. To support this claim, we have shown that our heuristic framework is correct in terms of being semantically equivalent to a given ranking function. In particular, we proved that all resulting domination relationships would also be effects of a ranking induced by some utility function with sensible characteristics such as being monotonic and composable. This actually means that all conceptual trade-offs resulting from our heuristic approach are qualitative representations of a more complex quantitative utility function used for ranking, while still avoiding the elicitation overhead generally imposed by quantitative approaches.

Finally, considering personalization issues our framework is especially valuable for handling the ever increasing amount of information handled by today's e-shopping platforms and market places. This is because trade-off skylines do not only allow for improving the quality of shopping result sets by removing suboptimal items, but additionally providing an semantic layer of personalization enabling users to quantitatively compensate between different types

of products as characterized by their typical attribute domain values. Our approach, thus, does not only improve the manageability of result sets, but also increases their focus and individual usefulness.

7. ACKNOWLEDGMENTS

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

[1] I.F. Ilyas, G. Beskales, and M.A. Soliman, "A survey of top-k query processing techniques in relational database systems," *ACM Computing Surveys*, vol. 40, 2008.

[2] S. Börzsönyi, D. Kossmann, and K. Stocker, "The Skyline Operator," *Int. Conf. on Data Engineering (ICDE)*, Heidelberg, Germany: 2001.

[3] D. Kossmann, F. Ramsak, and S. Rost, "Shooting stars in the sky: an online algorithm for skyline queries," *Int. Conf. on Very Large Data Bases (VLDB)*, Hongkong, China: 2002.

[4] S. Zhang, N. Mamoulis, and D.W. Cheung, "Scalable skyline computation using object-based space partitioning," *35th SIGMOD int. conference on Management of Data*, Providence, Rhode Island, USA: 2009.

[5] J. Lee and S.-won Hwan, "BSkyTree: scalable skyline computation using a balanced pivot selection," *13th Int. Conference on Extending Database Technology (EDBT)*, Lausanne, Switzerland: 2010.

[6] J. Selke, C. Lofi, and W.-T. Balke, "Highly Scalable Multiprocessing Algorithms for Preference-Based Database Retrieval," *15th International Conference on Database Systems for Advanced Applications (DASFAA)*, Tsukuba, Japan: 2010.

[7] T. Kim, J. Park, J. Kim, and H. Im, with S. Park, "Parallel skyline computation on multicore architectures," *25th International Conference on Data Engineering (ICDE)*, Shanghai, China: 2009.

[8] W.-T. Balke, J.X. Zheng, and 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.

[9] P. Godfrey, "Skyline cardinality for relational processing. How many vectors are maximal?," *Symp. on Foundations of Information and Knowledge Systems (FoIKS)*, Vienna, Austria: 2004.

[10] P. Fishburn, "Preference structures and their numerical representations," *Theoretical Computer Science*, vol. 217, Apr. 1999, pp. 359-383.

[11] C. Lofi, U. Güntzer, and W.-T. Balke, "Efficient Computation of Trade-Off Skylines," *13th International Conference on Extending Database Technology (EDBT)*, Lausanne, Switzerland: 2010.

[12] C. Lofi, W.-T. Balke, and U. Güntzer, "Consistency Check Algorithms for Multi-Dimensional Preference Trade-Offs," *International Journal of Computer Science & Applications (IJCSA)*, vol. 5, 2008, pp. 165-185.

[13] C. Lofi, W.-T. Balke, and U. Güntzer, "Efficient Skyline Refinement Using Trade-Offs," *3rd International IEEE Conference on Research Challenges in Information Science (RCIS)*, Fès, Morocco: 2009.

[14] P. Viappiani, B. Faltings, and P. Pu, "Preference-based search using example-critiquing with suggestions," *Journal of Artificial Intelligence Research*, vol. 27, 2007.

[15] R. Herbrich, T. Graepel, and K. Obermayer, *Large Margin Rank Boundaries for Ordinal Regression*, MIT Press, 2000.

[16] C. Domshlak and T. Joachims, "Efficient and non-parametric reasoning over user preferences," *User Modeling and User-Adapted Interaction*, vol. 17, 2007, pp. 41-69.

[17] T. Joachims and F. Radlinski, "Search Engines that Learn from Implicit Feedback," *IEEE Computer*, vol. 40, 2007, pp. 34-40.

[18] C. Lofi and W.-T. Balke, "On Skyline Queries and how to Choose from Pareto Sets," *Intelligent Query Processing*, Springer, 2011.

[19] J. Lee, S.-won Hwang, Z. Nie, and J.-R. Wen, "Product EntityCube: A Recommendation and Navigation System for Product Search," *6th IEEE International Conference on Data Engineering (ICDE)*, Long Beach, California, USA: 2010.