SkyView: a User Evaluation of the Skyline Operator

Matteo Magnani Dept. of Info. Technology, Uppsala University, Sweden matteo.magnani@it.uu.se Ira Assent Dept. of Computer Science, Aarhus University, Denmark ira@cs.au.dk Kasper Hornbæk Dept. of Computer Science, Uni. of Copenhagen, Denmark kash@diku.dk

Mikkel R. Jakobsen Dept. of Computer Science, Uni. of Copenhagen, Denmark mikkelrj@diku.dk Ken Friis Larsen Dept. of Computer Science, Uni. of Copenhagen, Denmark kflarsen@diku.dk

ABSTRACT

The skyline operator has recently emerged as an alternative to ranking queries. It retrieves a number of potential best options for arbitrary monotone preference functions. The success of this operator in the database community is based on the belief that users benefit from the limited effort required to specify skyline queries compared to, for instance, ranking. While application examples of the skyline operator exist, there is no principled analysis of its benefits and limitations in data retrieval tasks. Our study investigates the degree to which users understand skyline queries, how they specify query parameters and how they interact with skyline results made available in listings or map-based interfaces.

Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous

Keywords

Skyline; visualization; interface; user evaluation

1. INTRODUCTION

Since its introduction more than a decade ago [7], the skyline operator has enjoyed significant attention in the database research community. Skyline queries introduce a retrieval model that selects all optimal choices for any monotone preference function. Figure 1 shows a classical example about hotel booking based on *distance to the beach* and *price*, where any hotel providing a trade off between these two criteria that is not matched by any other hotel (termed "dominated") is part of the skyline.

This example highlights the intuitive semantics and minimal setup requirements of the skyline operator: no weights need to be specified for the two criteria. However, the data visualization presented in Figure 1 differs from typical hotel booking interfaces on real Internet sites, like expedia.com or

CIKM'13, Oct. 27–Nov. 1, 2013, San Francisco, CA, USA. Copyright 2013 ACM 978-1-4503-2263-8/13/10 ...\$15.00. http://dx.doi.org/10.1145/2505515.2505739 .



Figure 1: Hotel example: the skyline of hotels in attributes price (x-axis) and distance to beach (y-axis) consists of all best trade-offs (black points)

like the one we have created for this paper and shown in Figure 2. For example, in a real hotel booking task more than two attributes can be used to describe the hotels, such as price, star rating, distance from multiple points of interest or user reviews. This prevents a graphical representation of the data like in Figure 1 and a corresponding intuitive interpretation of the result. In addition, different users may want to choose different skyline attributes. This decision may seem trivial, but in practice it must be combined with other data management operators, like filtering, also involving the choice of some attributes. This combination can be confusing and the difference between selecting an attribute as a skyline or as a filter (or both) may not be easy to understand. Additionally, while providing support for skyline queries the interface should not be overloaded with options. Finally, data can be returned to the user through a map, by sorting it according to user preferences or by highlighting selected (representative) skyline hotels. These different visualizations can be integrated, but how understandable and practical would they be?

Following this discussion, while the advantages of the skyline operator model are easily accepted for multi-criteria

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Price and Location		g map round round round round gooddoor riond				
Price per Night:	Order by: Price per Night (lowest first)					
☑ Distance from Colosseum:	← Previor	Showing hotels 1-10 out of 1024 Next	;→			
	HIN A	Seven Hills Village				
Distance from Trevi Fountain:		Via Vittorio Trucchi 10, 00189 Rome Price per Night: 23€. 0 stars	ails			
Distance to the Vatican:		Quality Hotel Rouge et Noir Roma Via Cosimo de Giorgi, 8, 00158 Rome Price per Night: 44E, 4 stars	ails			
Facilities		Camping Village Fabulous Via Di Malafede 225, 00125 Rome Price per Night: 45E. 0 stars	ails			
Pool		Quality Hotel Excel Ciampino Airport				

Figure 2: SkyView, a typical hotel booking interface. SkyView and its extended support for skyline queries are described in details in Section 3

data, to the best of our knowledge no analysis exists of the benefits (and potential limitations) of the operator when real users are involved. Even more surprisingly, it has not been studied how existing interfaces should be extended to incorporate skyline queries, both with respect to the specification of queries and the presentation of the results. Some work has been done on applying the skyline operator in different contexts [1, 13, 6], but it has focused on the application domain or on diversification of results. In this work, we propose to analyze the skyline operator with respect to its applicability to data retrieval tasks, using as a paradigmatic example the most frequently discussed application scenario, namely that of booking a hotel. To this end, we introduce a visualization of a hotel booking web site (SkyView) that mimics systems used today, while seamlessly integrating the skyline functionality. Users are asked to perform hotel booking tasks in a think aloud study based on real hotel data of a large European city. We analyze their behavior and their results to obtain an understanding of the usefulness of skyline queries and of different options to integrate skyline queries in a user interface.

2. STATE OF THE ART

The **skyline operator** was proposed in [7] together with basic algorithms. Subsequent works have introduced algorithms based on sorting [9], indexing [22], and on the progressive [22] and anytime [16] query models.

Extensive studies on efficient skyline query execution have been complemented by several works questioning the direct usability of the skyline operator and proposing alternative query types. In particular, a well studied problem concerns the size of the skyline that can become too large, requiring an additional filter to provide users with a few representative records. This is usually called a **representative skyline** and constitutes the main advanced functionality integrated into SkyView to perform our user study. Several works, e.g., [21, 8, 26, 28], have approached this problem focusing on the data, e.g., they try to compute the relevance of a skyline record by checking its relationships with the other records. On the contrary, some approaches have focused on incorporating models of the users in the record selection algorithms, as described in the following.

Some approches are based on explicit assumptions about the **behavior and expectancy of users**, without exploiting any interaction with them. [19] assumes users can be described by linear utility functions while [10] describes a model based on step utility functions. In a slightly different context, [5] proposes to infer missing scoring functions for some users by looking at other similar users.

Other studies have proposed to rely on direct input from the users that can guide the selection of skyline records. The idea of asking for user feedback to guide the computation of the skyline is exploited in [3, 15], and some papers use user feedback to elicit preferences to be incorporated into the algorithms [2, 4, 29, 18]. In [14] users are required to specify a partial order over data dimensions, so that some combinations are first used to compute small skylines, and additional dimensions are included only if more records are required.

In all these works, abstract users are modeled and their preferences are used to guide the selection algorithms. However, to the best of our knowledge no systematic study involving real users has been reported in the literature. As a result, we currently know several advanced skyline-related methods but we do not know how real users would interact with even a simple skyline query, in which way they would specify it and to which extent they would understand its result. This paper aims at providing the first study in this direction.

3. THE SKYVIEW APPLICATION

The SkyView interface depicted in Figure 2 consists of two main parts: the query panel (left) containing controls with which the user can specify some requirements, and the results panel (right) showing the hotels matching the user's query. As the user manipulates the controls the results panel updates to show only the hotels matching the query.

We considered alternative approaches for allowing users to interact with the skyline operator: *explicit vs. implicit* selection of skyline criteria, *map vs. list* result visualization and *highlight vs. push-up* of representative skyline suggestions, all described in the following.

In the user study, participants used two interface variants implementing explicit or implicit criteria selection:

- Explicit selection: users explicitly select each skyline criterion, e.g., by clicking on a check box next to the *distance* attribute as in Figure 3(a).
- **Implicit selection:** the system infers the skyline criteria from the user's manipulations of the query controls, e.g., using *distance* as a skyline attribute when the user selects a distance range (Figure 3(b)).

Results could be visualized as in real on-line booking sites:

- Map view: the user can zoom and pan in map using the mouse (see Figure 3(c) and (d)). If the user clicks on a mark, a popup window is shown with information about the corresponding hotel.
- List view: shows information for each hotel matching the user's query, listed in the order chosen from a drop-down menu which includes name, price, distance, and rating (Figure 3(e)).

Skyline-based suggestions could be emphasized in two ways:

• **Highlight:** hotels in the representative skyline are distinguished using a yellow color/marker, but they are shown together with the other hotels matching the query (Figure 3(c)). When using a list, all hotels are ordered by the user-selected sorting criteria.



Figure 3: Various user interface elements: (a) explicit selection of a skyline attribute, (b) implicit selection (activated when the slider is operated), (c) map with highlighting, (d) map with push-up and (e) list with push-up

• **Push-up:** representative skyline hotels are shown as a distinct group. In the list view, they appear before the other hotels, similar to how widespread search interfaces show a group of sponsored or "featured" items before the actual results that match the criteria (Figure 3(e)). In the map view, they become the only visible hotels (Figure 3(b)).

The system is implemented as web-application with a frontend (UI) written in Javascript and HTML5 and the back-end written in Java. The front-end and the back-end communicate using a simple AJAX request-reply protocol transferring query results encoded in JSON. The system was deployed and tested using Google App Engine.

3.1 Data management

The data management operators available through our visual interface are: constraint-based filtering, sorting on single attribute, skyline and K-representative skyline.

Constraint-based filtering and sorting are the basic operators already available in all main on-line booking systems. For example, we can look for hotels that are cheaper than \notin 90 (this corresponding to a simple WHERE SQL clause) and sort the result by hotel price (this corresponding to a simple ORDER BY SQL clause). Sorting can be expressed only on a single attribute at a time, e.g., *price*, as in existing on-line hotel booking interfaces.

Once a set of attributes of interest has been specified, the **skyline** operator can be used to filter out some hotels. In particular, all hotels that are *dominated* (i.e., worse on at least one attribute and not better in any other) by some other hotel are discarded. As an example, consider the hotels indicated in Figure 4 and assume that a user is interested in cheap hotels close to the Colosseum and preferably having a pool. Skyline hotels are highlighted in boldface. *Spanish Steps View* is filtered out (not included in the skyline) because it is dominated by *Mercure*. In fact, *Mercure*

Name	Price (€)	Dist (km)	Pool
Seven Hills Village	23	13	Yes
The Strand	76	5.7	Yes
Spanish Steps View	1219	1.9	No
Mercure	240	.5	Yes
Happy Village	65	12.9	Yes
Turati	60	1	No
Rhome86	52	1.1	No

Figure 4: Hotel data. Boldface hotels are in the skyline. Skyline hotels with gray background constitute one possible representative skyline

is cheaper ($\in 240$ instead of $\in 1219$), it is closer to the Colosseum (.5 km instead of 1.9 km) and has a pool (not present in *Spanish Steps View*). Of course, *Spanish Steps View* could be selected in a skyline where different attributes are regarded as relevant.

Another important functionality implemented in our interface is the representative skyline, which enables the computation of a small set of skyline hotels to be suggested to the user. In Figure 4 is indicated a possible 4-representative skyline set (gray background). These hotels together provide an overview of the main types of hotels available in the skyline: from top to bottom, we have selected a hotel that is cheap but far from the Colosseum, a more expensive hotel that is at an average distance, a very close hotel which is very expensive, and finally a hotel that provides a good compromise between distance and price but has no swimming pool. Hotel *Turati* has not been selected because it is very similar to hotel *Rhome86*: users who would sacrifice the pool to favor the other two attributes would already find a good option in the representative skyline without needing to browse additional hotels. To compute a k-representative skyline we adopt a greedy algorithm, as in [26]. We refer the reader to this publication for the algorithmic details, that are not central with regard to our work — also in this case the limited size of the data, their well understood meaning and the direct control over the data do not require more sophisticated approaches.

In this work we use a database of 1024 hotels in **Rome**. Basic hotel information, including a picture, star rating, price of a room per night, location and textual description, has been extracted from a hotel booking website¹. From the location we have computed the distance from three points of interest (Trevi Fountain, Colosseum and the Vatican). From the description we have checked the presence of some facilities, e.g., Wi-Fi Internet connection. The final data consist of 16 attributes, organized in three classes: descriptive attributes, not used in the skyline computation but providing information that may be exploited by the users to choose the preferred hotel (name, address, city, postal code, short description, picture, latitude, longitude), numeric skyline attributes (star rating, trip advisor rating, price, distance from Vatican, distance from Colosseum, distance from Trevi fountain), and Boolean skyline attributes (pool, Internet).

These data correspond to the prototypical application scenario of skyline queries, as presented in [7]. The application domain is easily understandable and familiar to most people, and data are real, which prevents unrealistic data distributions from affecting the result of the queries. As a consequence, we expect user comments retrieved during our study to refer to the skyline operator itself, and not being biased by e.g. the choice of inappropriate data.

4. USER STUDY

The objective of the study was to get information on how users interact with alternative skyline-enhanced interfaces and how well they understand and take advantage of the representative skyline. The aim was not to measure task performance, so we decided against running a controlled experiment (as warned about in the literature [24]): while a quantitative study can be seen as one of the next steps in this research project, a principled performance study cannot be executed without a clear understanding of the preferred way to integrate the skyline operator in the user interface.

4.1 Tasks

We formulated three task types, each created in four variants (explicit/implicit selection and highlight/push-up suggestions). For each interface, tasks were given in increasing order of difficulty as recommended in [23]. Tasks were based on intuitions about what tasks users do on travel data, on previous studies of geo-tagged housing data [27], and on a taxonomy of tasks done with information visualizations [25]:

- Find a hotel that satisfies a certain criterion, e.g., "Find a hotel that is at least 10 kilometers from the Colosseum".
- Find a hotel that satisfies several criteria, e.g., "Find the most expensive hotel that is near the Trevi Fountain and that has Internet".
- Find a hotel that is near a map feature and that satisfies several criteria, e.g., "Find a hotel that is near the major road around Rome (E80) but still highly rated".

In addition to these tasks, we concluded the session with an open-ended task in which we asked participants to imagine they were travelling to Rome on vacation. We asked them to describe which criteria they would apply to find a hotel. We also asked which of the four interface variants they would prefer to use. After they had done so, they went on to complete the task. We concluded by asking about the extent to which they felt that the results matched their criteria.

4.2 Procedure

Six participants were recruited by word of mouth (two female). This number of participants is often recommended for formative user studies [20].

The procedure followed the one recommended in [23]. Participants were first welcomed and briefed on the purpose of the study. Because participants were video-recorded, they were asked to sign a form of consent. Then they were introduced to the think aloud protocol and briefly trained on how to think aloud. After that, we started the application with one of the four interfaces. We explained how the interface worked while allowing participants to try the interface and ask questions. Participants then solved the three tasks, which were handed to them on a piece of paper one at a time. When moving on to the next interface we explained the specific differences from the interface they had just used. We urged participants to think aloud if they fell silent and probed them to understand their intentions and how they interpreted the hotel suggestions. After completing all closed tasks we gave the open-ended task and elicited their overall experience with the interfaces and their preferences. Sessions lasted around 45 minutes on average.

During the study participants used a 24-inch display on which the interface was shown in a browser window 800x600 pixels in size, and a wireless mouse to control the interface. To collect the data we used a program to capture screen recording together with video of the participants while they completed the tasks. The results were analyzed by using the method called *instant data analysis* [12].

5. EXPERIMENTAL RESULTS

The results of the think-aloud study fall in three groups, which concern participants' understanding of the skyline operator, their reactions to the user interfaces (reported in this section) and ideas for improving the way the skyline operator may be integrated in user interfaces (presented in the final discussion).

5.1 Understanding the Skyline Operator

Participants understood the skyline operator in different ways. Although we provided only minimal explanation of how hotels were suggested from participants' preferences, at least two users spontaneously offered acceptable, lay explanations of the skyline operator. One noted that it did a *magic weighing*, another that it weighed the selected dimensions together to *maximally fulfill the criteria*. All participants answered tasks with one of the suggested hotels, and several participants made at least one choice *because the hotel was suggested*. In that regard, the SkyView user interface worked well.

However, the think aloud comments also suggest that all participants misinterpreted the working of the skyline operator on one or more occasions. This was most clear when

¹http://www.expedia.com

participants were asked to explain suggested hotels or to talk about the difference between hotels they chose as answers to the tasks and hotels suggested by the representative skyline operator. Responses included why this ranking? No clue or it is really hard to understand why something is suggested. Some participants even speculated that the skyline was based on their behavior in previous tasks or on a weighing together of both selected and non-selected parameters (one participant remarked that the skyline was *probably* based on prices, and distance to the Colosseum, and what do I know, although he had only set the number of stars for a hotel to be used for calculating the skyline). The misunderstanding of the skyline operator was also clear in participants' behavior. For instance, several participants would set a parameter for inclusion in the skyline and then also set a maximum price, even though this is unnecessary when only one parameter is chosen. At least one participant would also act as if he expected increasing the upper bound of a filter to result in new hotels being suggested, even though that is impossible when one parameter is being minimized.

The choice of limiting the representative skyline to five hotels had several consequences. When many hotels could be in the skyline, for instance because they shared a top rating, the choice to show five was arbitrary. Three participants commented on this. One participant was also confused that using just one parameter for the skyline could give five results: having selected the star rating parameter, he was not sure why a five-star hotel was shown among four-star hotels further down in the list and not together with the suggested five-star hotels near the top (due to the fact that non-suggested hotels were sorted according to another criterion, e.g., price). We question whether it makes sense to show skyline results for just one parameter (in which case results can simply be ordered). Several participants toggled a parameter on and off for the skyline and saw how the suggested hotels would change; one of them commented that this non-deterministic behavior, caused by the adopted optimization algorithm, was confusing.

In one case, a participant was confused with the fact that if filtering causes few results to be returned (in his case two), all of those results may be in the skyline. This may be formally correct, but was not interesting to the participant.

5.2 Understanding the User Interfaces

Some of the participants' behavior concerned to a larger extent the user interface, in particular the key choices of implicit vs. explicit selection of parameters for the skyline and of highlighting vs. push-up of selected hotels.

Participants encountered several problems with an explicit choice of parameters. At least four users missed clicking the checkbox for a parameter that they wanted to include in the skyline. Participants who did not click on a parameter to include it in the skyline held the misconception that the suggested hotels were minimized on that parameter. Clicking the checkbox can be considered an *additional step* that users might miss. Implicit setting of skyline parameters worked better, although one participant noted that they had to activate a filter to set a skyline preference (and thereby risk removing some hotels that could be of interest).

A key decision in the user interface design was to place the selection of parameters for the skyline in the query panel in the left part of the interface (see Figure 2). Many participants were confused about the difference between mechanisms for filtering, ordering results, and setting parameters for the skyline. Some participants would talk about skyline parameters as a particular way of ordering, others of it as a kind of filtering. These data suggest that perhaps participants find the consequences of filtering and ordering more straightforward to understand than adding parameters to an algorithm that suggests hotels. Thus it seems worthwhile to investigate a simpler mental model for participants about the skyline and the skyline-based suggestions.

Highlighting and push-up seemed to offer different benefits for the list and map views. Highlighting keeps any ordering that the user may have requested in the list view and avoids misunderstandings about the ordering. Unfortunately, participants did not see many of the suggested hotels when they were just highlighted: in cases where suggested hotels were not among the ten shown on the first page, the only indication that they were in the list of hotels was the text about the number of suggestions (which most participants seem to ignore). One participant using the combination of highlighting and implicit choices of parameters seemed unaware that suggestions were generated. Highlighting in the map view, where suggested hotels were easily distinguished from the others, did not have these drawbacks and was well received.

The specific implementation of push-up in the present interface using yellow highlighting and a position at the top of the list made participants see the suggestions. However, two participants commented that it looked like advertisements. Push-up in the map view, where only suggested hotels were shown, did not work well for five participants: one remarked that seeing just one recommendation on a map was "unsatisfactory", another was puzzled when no hotels were shown (since no skyline parameters were set).

6. DISCUSSION AND CONCLUSION

While the skyline and representative skyline are attractive, integrating them in a user interface so that they are understandable and useful for users is hard.

Here, it would be interesting to study some of the alternative models for skylines that have been discussed in the data management literature, such as thick skylines that provide more results than just the optimal hotels [11], aggregate skylines that return skyline groups [17] and deterministic representative skyline algorithms.

One interpretation of participants' comments is that the skyline should be more prominent in the user interface. One way of doing so would be to show why a particular hotel is being suggested. For instance, the interface could incorporate a textual explanation or small visualizations explaining the strengths of each proposal.

An alternative interpretation of the participants' comments is that the suggested hotels should be emphasized less in the interface. The comments of one participant in particular support this notion: he said that he expected the most relevant hotels to be listed first, similar to the ranking by search engines. The assumption here is that if the suggestions work well there is little reason to explain them. Based on the discussion about ordering vs. filtering above, one idea is to integrate skyline parameter setting with the order functionality (to the top right in Figure 1). That would provide a mental model of a skyline query as a way to order (at least in part) on several criteria simultaneously: a model that was mentioned by one of the participants when we asked him why some hotels were suggested. Another idea comes from observations of tasks where participants used map features that could not be specified as skyline parameters (such as finding a hotel near the water). The skyline could give suggestions from among only those hotels that are within the current map view. That would combine the benefit of free exploration on the map with the strength of having optimal suggestions for hotels to explore.

Finally, we would like to draw particular attention to the (expected) fact that many tasks were completed without actually choosing a hotel in the representative skyline. People may be more or less good at searching, they might be using subjective criteria such as the personal impression of a hotel based on the images provided, and criteria like being close to the water are not available through the interface.

Our work has partially supported the belief that using the skyline in actual user decision making tasks is a promising approach. It has also shown that skylines deserve further attention to establish the most successful strategies for explaining, parameterizing and comparing results. Although the think aloud study has identified interesting possibilities and challenges in integrating the skyline operator in user interfaces, we stress that our evaluation is only the first step in developing a full understanding of how (and if) users can benefit from using this operator.

Acknowledgement

This work has been supported in part by the Danish Council for Strategic Research, grant 10-092316, and Italian MIUR FIRB project RBFR107725.

7. REFERENCES

- Mohammad Alrifai, Dimitrios Skoutas, and Thomas Risse. Selecting skyline services for qos-based web service composition. In WWW, 2010. ACM.
- [2] Wolf-Tilo Balke, Ulrich Güntzer, and Christoph Lofi. Eliciting Matters – Controlling Skyline Sizes by Incremental Integration of User Preferences. In Advances in Databases: Concepts, Systems and Applications, volume 4443 of LNCS, 2007.
- [3] Wolf-Tilo Balke, Jason Zheng, and Ulrich Güntzer. Approaching the Efficient Frontier: Cooperative Database Retrieval Using High-Dimensional Skylines. In DASFAA, volume 3453 of LNCS, 2005.
- [4] W.T. Balke, U. Güntzer, and C. Lofi. User interaction support for incremental refinement of preference-based queries. In *RCIS*, 2007.
- [5] I Bartolini, Z. Zhang, and D Papadias. Collaborative Filtering with Personalized Skylines. *IEEE Trans. on Knowledge and Data Engineering*, 23(2), 2011.
- [6] Christian Beecks, Ira Assent, and Thomas Seidl. Content-based multimedia retrieval in the presence of unknown user preferences. In *MMM*, 2011.
- [7] S Börzsönyi, D Kossmann, and K Stocker. The Skyline operator. In *ICDE*, 2001.
- [8] Chee-Yong Chan, H V Jagadish, Kian-Lee Tan, Anthony K H Tung, and Zhenjie Zhang. On High Dimensional Skylines. In *EDBT*, 2006.
- [9] J Chomicki, P Godfrey, J Gryz, and D Liang. Skyline with presorting. In *ICDE*, 2003.
- [10] Atish Das Sarma, Ashwin Lall, Danupon Nanongkai, Richard J. Lipton, and Jim Xu. Representative

skylines using threshold-based preference distributions. In *ICDE*, 2011.

- [11] Wen Jin, Jiawei Han, and Martin Ester. Mining thick skylines over large databases. In *PKDD*, 2004.
- [12] Jesper Kjeldskov, Mikael B. Skov, and Jan Stage. Instant data analysis: conducting usability evaluations in a day. In *NordiCHI*, 2004.
- [13] H.-P. Kriegel, M. Renz, and M. Schubert. Route skyline queries: A multi-preference path planning approach. In *ICDE*, 2010.
- [14] Jongwuk Lee, Gae-won You, and Seung-won Hwang. Personalized top-k skyline queries in high-dimensional space. *Information Systems*, 34(1), 2009.
- [15] Christoph Lofi, Ulrich Güntzer, and Wolf-Tilo Balke. Efficient computation of trade-off skylines. In *EDBT*, 2010.
- [16] Matteo Magnani, Ira Assent and Michael L. Mortensen. Anytime skyline query processing for interactive systems. In *DBRank*, 2012.
- [17] Matteo Magnani and Ira Assent. From stars to galaxies: skyline queries on aggregate data. In *EDBT*, 2013.
- [18] Denis Mindolin and Jan Chomicki. Preference elicitation in prioritized skyline queries. *The VLDB Journal*, 20(2), 2011.
- [19] Danupon Nanongkai, Atish Das Sarma, Ashwin Lall, Richard J. Lipton, and Jun Xu. Regret-minimizing representative databases. *PVLDB*, 3(1-2), 2010.
- [20] Jakob Nielsen and Thomas K. Landauer. A mathematical model of the finding of usability problems. In *CHI*, 1993.
- [21] D Papadias, Y Tao, G Fu, and B Seeger. An optimal and progressive algorithm for skyline queries. In SIGMOD, 2003.
- [22] Dimitris Papadias, Yufei Tao, Greg Fu, and Bernhard Seeger. Progressive skyline computation in database systems. ACM TODS, 30(1), 2005.
- [23] Jeffrey Rubin. Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests. John Wiley & Sons, Inc., New York, NY, USA, 1994.
- [24] W. R. Shadish, T. D. Cook, and D. T. Campbell. Experimental and Quasi-experimental Designs for Generalized Causal Inference. Houghton Mifflin, Boston, MA, USA, 2002.
- [25] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In VL, 1996.
- [26] Y Tao, L Ding, X Lin, and J Pei. Distance-Based Representative Skyline. In *ICDE*, 2009.
- [27] C Williamson and B Shneiderman. The dynamic homefinder: evaluating dynamic queries in a real-estate information exploration system. In *SIGIR*, 1992.
- [28] Z Zhang, H Lu, B C Ooi, and A K H Tung. Understanding the meaning of a shifted sky: a general framework on extending skyline query. *The VLDB Journal*, 19(2), 2010.
- [29] F Zhao, G Das, K L Tan, and A K H Tung. Call to order: a hierarchical browsing approach to eliciting users' preference. In *SIGMOD*, 2010.