OLAP Preferences: A Research Agenda

Stefano Rizzi
DEIS - University of Bologna
Viale Risorgimento, 2
Bologna, Italy
srizzi@deis.unibo.it

ABSTRACT

Expressing preferences when querying databases is a natural way to avoid empty results and information flooding, and in general to rank results so that the user may first see the data that better match his tastes. In this paper we outline the main research issues to be faced in order to develop a system for handling user preferences on OLAP cubes.

Categories and Subject Descriptors

H.4.2 [**Information Systems Applications**]: Types of Systems—*Decision support*; H.3.3 [**Information storage and retrieval**]: Information search and retrieval—*query formulation*

General Terms

Management

Keywords

User preferences, data warehouses, OLAP

1. INTRODUCTION AND MOTIVATION

Personalizing e-services by allowing users to express preferences is becoming more and more common. When querying, expressing preferences is seen as a natural way to avoid empty results on the one hand, information flooding on the other. Besides, preferences allow for ranking query results so that the user may first see the data that best match his tastes.

Though a lot of research has been carried out during the last few years on preferences in databases, no attempt to develop a theory of preferences on OLAP (multidimensional) cubes has been made yet. On the other hand, cubes (implemented either on relational or on multidimensional platforms) are the core of data warehousing systems and business intelligence systems. Their users are decision-makers who need to express complex queries through OLAP front-end tools, often returning huge volumes of data, sometimes returning little or no information. Thus, we argue that expressing preferences could be highly valuable in this domain.

Consider for instance a classical marketing scenario, where users covering different roles ask for accessing data in different contexts, through devices with different computation and visualization capabilities (PCs, palmtops, mobile phones), and with customized presentations. The sales manager is mainly interested in monthly revenues, but he may also wish to see more detailed data if the weekly revenue exceeds 10 k Euro. A sales agent prefers to see highest commissions data for his own customers, while a branch manager is rather more interested in products of a given category selling abnormally low quantities. Formulating such queries with classical OLAP interfaces, i.e., by expressing *hard constraints* on the cube data, would probably lead either to information flooding, which is particularly critical when working with devices with limited visualization capabilities, or to empty results. Conversely, formulating the same queries by expressing preferences, meant as *soft constraints* on data, could improve the user satisfaction by returning all (and, possibly, only) the information that achieves the best compromise between his wishes.

In this paper we outline the main research issues to be faced in order to develop a system for handling user preferences on OLAP cubes, with specific reference to the problems related to the preference model, to context-awareness, to query processing and optimization, and to user interfaces.

2. BASICS ON PREFERENCES

A major classification of preferences distinguishes between *quantitative* and *qualitative* preferences. While the first are indirectly expressed by means of a scoring function that associates a numerical score to each tuple returned by a query, the second are directly expressed as binary relations on the space of tuples. In the following we will focus on qualitative preferences, that yield higher expressiveness than quantitative ones.

From a theoretical point of view, the two leading approaches for dealing with qualitative preferences on relational databases are due to Kießling [5] and Chomicki [3]. Both define a preference as a strict partial order (s.p.o.)[1] over the set of possible tuples, i.e., the set of all possible combinations of attribute values, since the properties of s.p.o.'s match the common intuition of a preference. Nevertheless, there is a substantial difference in the way preferences are declared and combined. In [5], complex preferences are inductively constructed by applying composition operators to a set of predefined *base preference constructors*, thus obtaining a preference algebra. Conversely, in [3] preferences are "freely" defined by formulating first-order *preference formulas*.

[1] A s.p.o. is an irreflexive and transitive (thus asymmetric) binary relation on the elements of a set [3].

DOLAP'07, November 9, 2007, Lisboa, Portugal.
ACM 978-1-59593-827-5/07/0011.

As concerns preference formulation, proposals have been made for extending SQL, like in [6].

Finally, as to preference processing and optimization, most proposals have been focusing on *best matches only* queries, that only return the tuples that are not dominated by other tuples according to the preference relation (e.g., [6, 7]), and on the subclass of *skyline queries*, where the preference relation is expressed as a Pareto composition of total orders on numerical attributes (e.g., [1, 2]).

3. A RESEARCH AGENDA

Applying in the OLAP context one of the approaches devised for handling preferences in relational databases is possible: after all, each (elemental or aggregated) event stored in a cube can be seen as a "flat" relational tuple including all dimensions, hierarchy levels, and measures. On the other hand, this naive solution would be unsatisfying for two reasons: (1) it would have low expressiveness, since it does not consider the peculiar characteristics of the multidimensional model; and (2) it would perform badly, since aggregation makes preference evaluation quite complex.

Thus, we argue that an ad hoc approach must be devised for dealing with preferences on OLAP cubes. A number of research issues must then be faced, related to both the preference model and its implementation. The main open issues are tentatively listed below:

- **Preference model**. The first problem to be faced in devising a preference model for OLAP is how to take aggregation into account. In fact, the aggregation level has a strong impact on the size of the result returned to the user, and its inappropriate setting may end in either obtaining very coarse, useless information or being flooded by tons of detailed data. For this reason we argue that, in the OLAP domain, users must be enabled to express their preferences on the query aggregation level too, for instance by stating that monthly data are preferred to yearly and daily data. Remarkably, following this approach would lead to allowing preferences on schema – rather than on instances as commonly done – to be expressed.

- **Context-awareness.** Context-awareness has often been coupled with user preferences (like in [8, 9]), in order to introduce a dependency between the user's wishes and the particular conditions he is operating in. Within a context-aware preference system, in an OLAP ubiquitous setting, the context could be determined by a set of dimensions such as the user role, the type of device he is operating, the type of information he wants to analyze, and his spatio-temporal location.

- **Query optimization.** In presence of complex preferences, it becomes essential to rely on a set of equivalence rules to be used for rewriting expressions in a better performing way. Though some work in this direction has already been done in pair with the above mentioned approaches, introducing operators for expressing preferences on the aggregation level will open new optimization possibilities.

- **Query processing.** In principle, preference queries on data warehouses could be processed by first calculating the whole data cube [4], then applying the optimization techniques devised for classical preference queries. Of course, given the huge size of the data cube, this would result in extremely poor performance. Thus, it is necessary to develop original processing techniques capable of efficiently coping with preferences on aggregation levels by relying on ad hoc algorithms and, possibly, new types of indexes.

- **User interface.** The first problem arising here is how to enable users to express preferences through an OLAP front-end. In fact, while in relational databases preferences are mostly expressed by defining extensions to SQL, querying of data warehouses is commonly carried out in a friendly, visual way by means of sophisticated graphical interfaces. A second problem is related to visualization of results. In fact, if preferences are to be also expressed on the aggregation level, data with different granularities may be returned together as the result of a query, which makes the traditional tabular and diagrammatic forms for viewing results fall short.

In conclusion, in order to enable OLAP users to express preference queries while preserving good performances, an original and comprehensive approach has to be devised entailing both theoretical and applied research on issues related to modeling, algorithms, languages, interfaces.

4. REFERENCES

[1] W.-T. Balke and U. Güntzer. Efficient skyline queries under weak pareto dominance. In *Proc. PREFERENCE*, Edinburgh, Scotland, 2005.

[2] I. Bartolini, P. Ciaccia, and M. Patella. SaLSa: Computing the skyline without scanning the whole sky. In *Proc. CIKM*, pages 405–414, Arlington, Virginia, 2006.

[3] J. Chomicki. Preference formulas in relational queries. *ACM Trans. on Database Systems*, 28(4):427–466, 2003.

[4] J. Gray, A. Bosworth, A. Layman, and H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-total. In *Proc. ICDE*, pages 152–159, New Orleans, Louisiana, 1996.

[5] W. Kießling. Foundations of preferences in database systems. In *Proc. VLDB*, pages 311–322, Hong Kong, China, 2002.

[6] W. Kießling and G. Köstler. Preference SQL - design, implementation, experiences. In *Proc. VLDB*, pages 990–1001, Hong Kong, China, 2002.

[7] T. Preisinger, W. Kießling, and M. Endres. The BNL++ algorithm for evaluating Pareto preference queries. In *Proc. PREFERENCE*, Riva del Garda, Italy, 2006.

[8] K. Stefanidis, E. Pitoura, and P. Vassiliadis. Adding context to preferences. In *Proc. ICDE*, pages 846–855, Istanbul, Turkey, 2007.

[9] A. H. van Bunningen, L. Feng, and P. M. G. Apers. A context-aware preference model for database querying in an ambient intelligent environment. In *Proc. DEXA*, pages 33–43, Kraków, Poland, 2006.