Specification-Based Data Reduction in Dimensional Data Warehouses

Janne Skyt Christian S. Jensen Torben Bach Pedersen
Department of Computer Science, Aalborg University
{skyt | csj | tbp}@cs.auc.dk

Data warehouses are often very large, with new data being appended regularly. The sheer size and complexity of such warehouses makes them hard to manage effectively and to query with the desired efficiency.

Although disk prices drop continuously, this neither solves the complexity nor the performance problems. Also, some data becomes unreliable or irrelevant as time passes. Examples include exact addresses of graduated students and the URLs of web pages. However, much such data continues to be of interest at an aggregated level as it ages.

This poster presents a powerful and easy-to-use technique [1] for aggregation-based data reduction that enables the gradual change of the data from being detailed to being increasingly aggregated. The technique enables huge storage gains while retaining the data that is essential to the users, and it preserves the ability to query original and reduced data in an integrated manner.

To illustrate the technique, assume a simple multidimensional data warehouse of an Internet Service Provider (ISP). The warehouse concerns click facts that are described by the dimensions *URL*, *time*, *user*, *session*, and *costumer_value_class*. Thus, the *granularity* of the facts is *URL* by *time* by *user* by *session* by *costumer_value_class*. The original URLs of clicks become uninteresting as clicks age because of web-site changes. This may cause the ISP to omit the detailed URLs for older data.

We consider the *URL* and *time* dimensions in more detail. In the *URL* dimension, *URL* values are contained in *domain* values, which are contained in *domain_group* values, which in turn are contained in a single top-value: *URL* $\leq$ *domain* $\leq$ *domain_group* $\leq \top_{URL}$. Similarly, the *time* dimension hierarchy is specified as follows: *day* $\leq$ *week* $\leq \top_{Time}$ and *day* $\leq$ *month* $\leq$ *quarter* $\leq$ *year* $\leq \top_{Time}$.

The data reduction technique is based on user-defined specifications consisting of a predicate that selects the facts to be reduced and a granularity to which these facts must be aggregated. An example could be: "Aggregate facts in the `.com` *domain_group* that are older than 12 months to the granularity *month* by *domain*." Higher aggregation levels can be specified similarly for even older data, which would cause a stepwise increasing aggregation of facts as they age.

To be semantically meaningful, a set of data reduction specifications must be *NonCrossing* and *Growing*. The former property requires that if two specifications select overlapping sets of facts, one of the specifications must consistently aggregate higher than the other in all dimensions. As a result there is always a well-defined lowest granularity for all facts, and the granularity of a fact is given by one single specification. The definition of this property contends with non-linear hierarchies. Specifically, two specifications aggregating into parallel hierarchies will always be crossing. Next, the *Growing* property ensures a gradually increasing aggregation of facts, so that a fact aggregated to some granularity will continue to be aggregated to at least that granularity. This reflects the irreversible nature of aggregation.

When querying reduced facts, varying granularities affect both selection and aggregation. For selection, the problem is that some facts may satisfy predicates partly, while for other facts, it cannot be determined if the predicates are satisfied. Three approaches may be taken to address this. In the conservative approach only facts that are known to satisfy the predicates are selected. For aggregation, the problem is that facts can have not only lower, but also higher granularities than asked for in a query. Four approaches may be taken to address this. In the so-called availability approach, all facts satisfying the predicate are included in the answer, and facts retain their granularity if this is higher than required and are otherwise aggregated to the required granularity.

Facts with varying granularities may be implemented using a set of disjoint subcubes, each of which contains facts of uniform granularity. When a set of facts are aggregated to a new, higher granularity, they are removed from their subcube, and the resulting, aggregated facts are stored in another subcube. Algorithms for moving aggregated facts and for querying a set of subcubes have been provided [1].

References

[1] J. Skyt, C. S. Jensen, and T. B. Pedersen. Specification-Based Data Reduction in Dimensional Data Warehouses. TIMECENTER Report TR-61, `www.cs.auc.dk/TimeCenter/`, July 2001.

Proceedings of the 18th International Conference on Data Engineering (ICDE'02)
1063-6382/02 $17.00 © 2002 **IEEE**