SUMMARIZABILITY

Arie Shoshani
Lawrence Berkeley National Laboratory, Berkeley, California
http://sdm.lbl.gov/~arie

SYNONYMS

Summarization correctness, statistical correctness

DEFINITION

Summarizability is a property that assures the correctness of summary operations over On-Line Analytical Processing (OLAP) databases, which are akin to Statistical Databases [5]. Such databases are generally referred to as "summary databases". Such databases have a data model based on one or more measures defined over the cross product of dimensions. For example, a book store company may have multiple stores in many cities. Assume that there is a database containing the stores revenues for books sold per day over the last three years. In such a database, "revenue" is a measure, and "book", "store", "day" are the dimensions that define the cross product over which the measure revenue is defined. A dimension in a summary database is said to be *summarizable* relative to a measure, if a summary statistic (sum, average, etc.) applied over the dimension produces correct results. For example, if summarization over all the books sold to obtain "total_revenues per store, per day" yields correct results, the dimension "book" is considered summarizable relative to the measure "revenue". Often dimensions are organized into a hierarchy of categories. For example, days can naturally be organized into months, and months into years. Similarly, books can be organized by book-types (e.g. cooking, fiction, etc.) Summarization can then be applied to categories, such as summarizing over books and days to get "revenue per book-type, per store, per year". In such cases the summarizability property must apply to each category level of the category hierarchy of a dimension, for that dimension to be considered summarizable.

HISTORICAL BACKGROUND

Statistical Databases, which were introduced in the 1980's [1], and OLAP databases, introduced in the 1990's [2, 3, 4] have a similar data model [5], but the issue of summarizability was not introduced until 1990 [6] and studied carefully until 1997 [7]. After that time, several authors have treated summarizability formally [8, 9, 10].

SCIENTIFIC FUNDAMENTALS

Next, examples that violate summarizability are presented, and using these examples the conditions for summarizability are stated. There are three such conditions that must hold in order for a dimension to be summarizable. Two of the conditions refer to the category levels in a dimension, and the third is a condition of the dimension relative to the measure. Next, some basic notation used throughout this document are introduced.

Notation

Consider, again, the example above of revenues per book, store, and day. Using a notation commonly used for such databases, this example database can be represented as "revenue (book, store, day)". For the category hierarchies of dimensions, the notation [C_1 -> C_2 -> … C_i -> C_{i+1}, … C_n] is used to represent a category hierarchy of a dimension of height n, starting from the more

detailed level towards higher levels. Thus, for the example above, where the two hierarchies mentioned above are over the dimensions book and day, the database will be represented as:

"revenue ([book-> book-type], store, [day -> month -> year])".

These concepts and notation are shown graphically in Figure 1, where the letters M, C, and X represent Measure, Category-level, and X-product (cross-product), respectively.

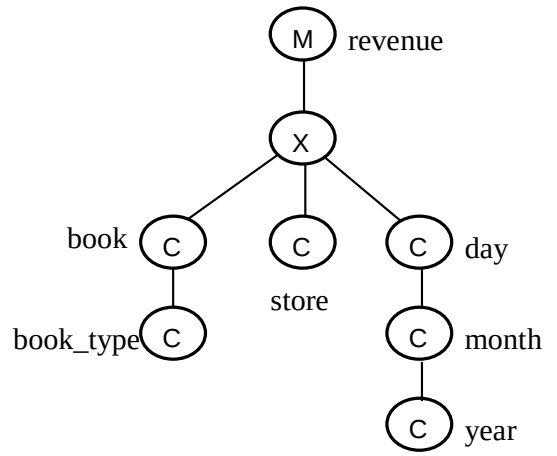

Figure 1: revenue of books sold in a particular store on a particular day

The "disjointness" condition

Consider the revenue database above. Suppose that the book-type set is (cooking, fiction, adventure, science, …). Most of the books will usually belong to a single book_type, but some could be categorized under two or more types. This is shown graphically in Figure 2, for sales in a particular day for particular store. As can be seen there in one book, b4, which is classified under the categories "fiction" and "adventure". If the revenues by book_type are added to generate "revenue (book-type, store, day)", the totals will seem to be correct. However, they are incorrect, because if the revenues for all book-types are added, the revenue for book b4 is counted twice. The reason is, of course, that book b4 belongs to two parent categories.

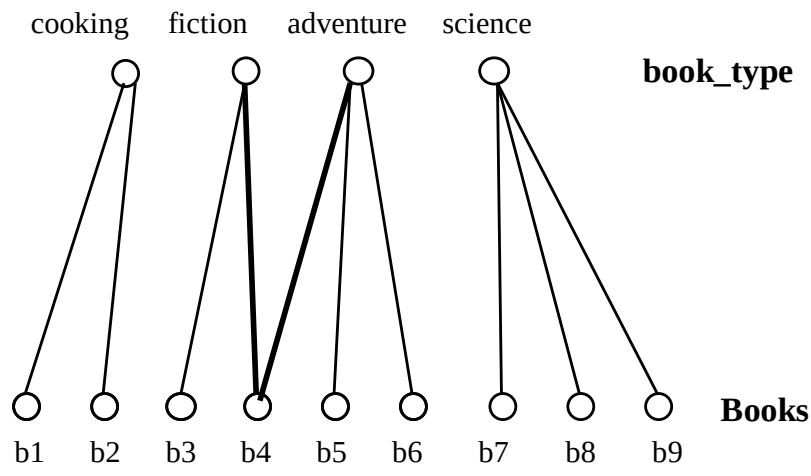

Figure 2: Books organized by book_type

The disjointness condition states: given two consecutive category-levels C_low and C_high in dimension, where [C_low -> C_high], the sets of lower-level categories elements that belong to each category element of the higher-level, must be disjoint. This condition can also be expressed as requiring that the category elements of the category levels form a "strict" hierarchy [8]. Yet another way to state this condition is to say that there must be a one-to-many relationship from C_high to C_low.

This seemingly simple observation is the source of incorrect statistics in many systems that do not enforce summarizability conditions. Under such conditions, summarization is still possible by special treatment, such as choosing to assign revenues equally to shared nodes (in the example above assigning ½ the revenue for book b4 to each of the "fiction" and "adventure" book types). However, this is not usually done. Note that summarization of book revenues to get, for example, "revenues (store, day) will yield a correct result, since the category book_type is not involved.

The "completeness" condition

Completeness is a condition that holds if all the children of higher-level elements exist. If some of the children are missing, then the summary to the higher level may be incorrect. Consider, for example, a database that contains "population (city, year, race, sex)". Suppose further that cities are organized by states, as shown in Figure 3. In this database, if the population is summarized to the "state" level, the result of populations is obviously incorrect, since only populations of cities are taken into account and not populations of villages and small towns. However, if the measure was stated as "populations_in_cities" then the [city -> state] category mapping would be summarizable. This example shows that the second condition of completeness is relative to the measure semantics.

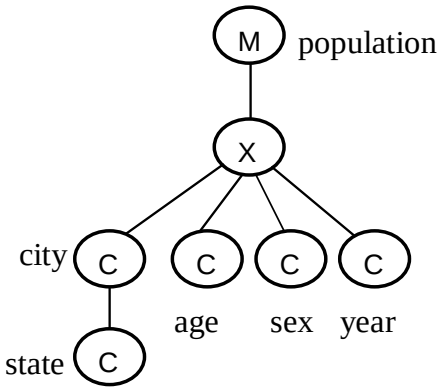

Figure 3: The population database is not summarizable to the "state" level

One way to overcome the "incompleteness" condition is to add instances that account for the missing elements in the category. In the population example, one can add for each state, in addition to the cities in the state, an instance that accounts for the population in all areas other than cities. Such a node can be labeled "other_areas" for example. If this is done, summarization to the state level would yield the correct summary population.

Another way to determine if a category level satisfies the completeness condition can often be based on external knowledge. For example, suppose that the dimension "age" in Figure 3 is organized into age_groups: (0, 10) (10, 20), …, (90-100). Is summarization to the age_group level correct in this case? It is only correct if there is external knowledge that this database does not contain people older that 100. Otherwise, a category (> 100) has to be added in order to satisfy the completeness condition.

The "measure type" condition

Consider the database in Figure 3 again, where a higher category level is added to the dimension year: [year -> decade]. Obviously, population cannot be summarized to the decade level, since adding the yearly populations does not yield a meaningful measure for the decade. However, if the measure was "average population" per year (and "counts" were also recorded), the average population per decade could be calculated. Why is that? As another example, consider the book revenues database in Figure 1. Obviously, the revenues can be summarized from days, to months, to years. However, if the measure was "number_of_unsold _books", this cannot be summarized (added) over the time dimension. The reason stems from the semantic behavior of "temporal aggregation".

In statistics the term “temporal aggregation” is used to describe the behavior of measures when aggregating over the time domain. The measures are classified into three types: “stock”, “flow”, and “value-per-unit”. It turns out that these types behave differently when summarized over time, depending on the summary statistics used. In particular, a measure of “stock” type cannot be summed over the time domain, whereas a measure of “flow” type can be summed over the time domain. In the example discussed above, “population” is of type stock, and so is “number_of_unsold _books”, and therefore they cannot be summed over the time dimension. In contrast, “book_revenues” is of type “flow”, and therefore can be summed.

In general, measures of type “stock” refer to a state of the measure recorded at a particular point in time (such as inventory), while measures of type “flow” record values of events over a period of a time (such as sales). A measure of the type “value-per-unit” is similar to “stock” in that it is recorded at a particular point in time, but it has a per-unit value (such as the cost of a book). In [7] a table is given for temporal summarizability for each measure type for five common aggregation operators: min, max, sum, avg, and range. The table is reproduced in figure 4. As can be seem only the operator “sum” is not summarizable for the types “stock” and “value-per-unit”. It turns out that “value-per-unit” is also not summarizable for non-temporal aggregation in the case of “sum”, but all other cases are summarizable for non-temporal aggregation [7].

	stock	flow	value/unit
min	yes	yes	yes
max	yes	yes	yes
sum	no	yes	no
avg	yes	yes	yes
range	yes	yes	yes

Figure 4: Temporal summarizability by measure type and function type

Summary

The conditions that are necessary to insure correctness of aggregation operations over statistical and OLAP databases were presented. Such conditions are referred to as “summarizability conditions”. Summarizability conditions apply to the dimensions of multidimensional data structures and to the category hierarchies in each dimension. Such conditions depend on the summary measure type, and whether the summarization is over the time domain. Note that the summarizability of each category level in a hierarchy is independent of the others; that is, some category levels can be summarizable while others are not.

Summarizability conditions are applicable for any database system that supports aggregation operations. While these conditions were described here in the context of the OLAP data model, these conditions apply to other models that do not express the multidimensional structures explicitly. In particular, it is possible to represent an OLAP schema as a relational schema, where the semantics of multidimensionality and category hierarchies are not explicit. In order to achieve correct results in aggregation operations from such relational databases, it is necessary to identify these (multidimensionality and category hierarchies) structures, and to make sure that the summarizability conditions hold. If all the conditions do not hold, aggregation operations should be avoided and/or refused.

KEY APPLICATIONS

It is often claimed that statistical operations can be inaccurate for various reasons. The better-known reason is summarization over null values. For example, taking an average over a set of values where some are null, will produce the wrong result if the null elements are represented as zeros, or if they are not discounted in the computation. Summarizability conditions are just as important, but are more subtle, semantically based, and are often overlooked. It is essential that such conditions are checked in any database that provides aggregation operators, including OLAP and relational database systems.

FUTURE DIRECTIONS

The three conditions described above for ensuring summarizability are necessary conditions. However, while it is believed that these conditions are also sufficient, this was not shown formally so far. Another aspect of future work is adding annotations the schemas as to whether summarizability conditions hold, and how to automate the checking of summarizability conditions dynamically in a database as data instances are entered into (or modified in) the database. Once summarizability is made part of the data model, it is necessary to enhance the aggregation operators to avoid summarization over non-summarizable data.

CROSS REFERENCES

Quality in Data Warehouses; Query Processing in Data Warehouses; Multidimensional Modeling; Scientific Databases; On-Line Analytical Processing; Dimension; Measure; Hierarchy.

RECOMMENDED READING

[1] P. Chan and A. Shoshani. Subject: A directory driven system for organizing and accessing large statistical databases. Conference on Very Large Data Bases, 1981, 553–563.

[2] E. F. Codd, S. B. Codd, and C. T. Salley. Providing olap (online analytical processing) to user-analysts: An IT mandate, E.F. Codd and Associates Technical report, 1993.

[3] J. Gray, A. Bosworth, A. Layman, and H. Pirahesh, Data cube: a relational aggregation operator generalizing group-by, cross-tabs and sub-totals. In Proceedings of 12th International Conference on Data Engineering (ICDE) 1996, 152-159.

[4] R. Agrawal, A. Gupta, and S. Sarawagi, Modeling multidimensional databases, in Proceedings of the 13th International Conference on Data Engineering (ICDE) 1997, 232-243.

[5] A. Shoshani, OLAP and statistical databases: Similarities and differences, in Proceedings of ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS) 1997, 185-196.

[6] M. Rafanelli, A. Shoshani, STORM: A Statistical Object Representation Model, Proceedings of the International Conference on Scientific and Statistical Database Management (SSDBM) 1990, 14-29.

[7] H-J. Lenz, A. Shoshani: Summarizability in OLAP and Statistical Data Bases, Proceedings of the International Conference on Scientific and Statistical Database Management (SSDBM) 1997, 132-143.

[8] T.B. Pedersen, C. S. Jensen, Multidimensional Data Modeling for Complex Data, Proceedings of the International Conference on Data Engineering (ICDE) 1999, 336-345.

[9] C. A. Hurtado, A. O. Mendelzon, Reasoning about Summarizability in Heterogeneous Multidimensional Schemas, International Conference on Database Theory (ICDT) 2001, 375-389.

[10] C. A. Hurtado, C. Gutiérrez, A. O. Mendelzon: Capturing summarizability with integrity constraints in OLAP, ACM Trans. Database Systems (TODS) 30(3), 2005, 854-886.