INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN 2278 – 0211 (Online)
ISSN 2278 – 7631 (Print)

Computation of Data Cube: A Data Mining Technique

Dr. Girish Katkar
Arts, Science and Commerce College, Koradi, Nagpur, India
Amit Dipchandji Kasliwal
Malegaon, Nasik, Maharashtra, India

Abstract:
This paper work is aimed to give a comprehensive view about the links between computation regarding data hierarchy through data cubes and data mining. The paper discusses the possibilities of hierarchical data computation, data mining and knowledge discovery. In this paper data mining algorithms are analyzed and the extracted knowledge is represented using On Line Analytical Processing cubes systems. The performance of the computational techniques and the ability to interpret it is have taken with vital importance. Also the resulting proposals have presented to be easy to understand. This is where the data cubes computational techniques come into picture as they satisfy the today's requirements.

Key words*: Data Mining, Data Cube, Attribute, Hierarchy, On Line Analytical Processing (OLAP)*

1. Introduction

In today's world there is a keep increasing amount of data, hence there must be some computational techniques for extracting the useful data from this vast quantity of data. But most of the data that we have is in the form of unstructured format as the some processes themselves do not have proper definitions to recognize that. Hence we need a computational system that can use this incomplete information. The guiding principle for data mining is the challenge to expand the large amount of variety of data in usefully manner so that it can be taken for consideration for requirement. In data mining, to manipulate the large amount of data, it has carious methods such as artificial neural networks, Decision trees, and the nearest-neighbor method and so on. Regardless of the technique used, the real value behind data mining is modeling is the process of building a model based on user-specified criteria from already captured data. The already captured data may have the some hierarchy and that hierarchy provides the data in unstructured as well as in "what-if" situations. So the data hierarchy is an arrangement of data consisting of sets and subsets such that every subset of a set is of lower rank than the set. Simple suggested Data Hierarchy refers to the systematic organization of data, often in a hierarchical form. That data organization involves fields, records, files and so on. So by computing (by mining) the data to find out the solution for a situation it may solve the problem. The techniques used for finding particular reason and with particular search we aim to apply techniques to data mining.

The distinction between data, information, knowledge and wisdom is foundational. An almost universally accepted view of these relationships was articulated by Stephen Tuthill at 3M in 1990.

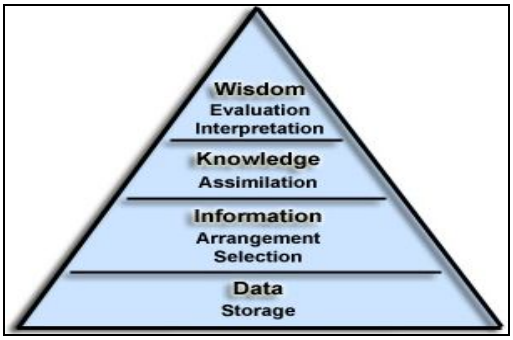

Figure 1

2. Online Analytical Processing Cube

An online analytical processing (OLAP) is a computer-based technique for analyzing large variety of data in the search for business intelligence. An OLAP cube is an array of data understood in terms of its 0 or more dimensions. A cube can be considered as a generalization of a three-dimensional spreadsheet. Cube is can be considered as shortcut for multidimensional dataset, given that data can have an arbitrary number of dimensions. The term hypercube is sometimes used, especially for data with more than three dimensions. For example, a company might wish to summarize financial data by product, by time-period, and by city to compare actual and budget expenses. Product, time, city and scenario (actual and budget) are the data's dimensions. The goal is to retrieve the decision support information from the data hierarchy with the help of cube in the most efficient way possible.

Data cubes are inherently multidimensional, with each dimension modeled with a hierarchy defining the levels of detail to use when aggregating the base fact table. In the simple case, these hierarchies are simple, uniform, and non branching so that there is only a single way to define the levels of detail for any particular dimension, i.e., a single path along that dimension. This type of data is commonly modeled using a star schema and is the type of data we have used in this paper to show how users can independently zoom on multiple hierarchies within a single visualization by associating hierarchies with the axes of a visualization.

3. Representation and Operations on Data Cubes in Data Hierarchy

Not only are data cubes widely used, but they also provide a powerful mechanism for performing data abstraction that we can as per requirements and for future purpose. A data cube is constructed from a subset of attributes in the database or from the data hierarchy. Certain attributes are chosen to be measure attributes, i.e., the attributes whose values are of interest. Other attributes are selected as dimensions or functional attributes. The measure attributes are aggregated according to the dimensions. In data Hierarchy, if it is represented in the form of above example the tables are two-dimensional representations of data, using columns and rows. Data cubes are multi-dimensional extensions of two-dimensional tables. A data cube can be thought of as a set of similar 2D tables stacked on top of each other. Data cubes categorize information into two classes: dimensions and measures, corresponding to the independent and dependent variables, respectively.

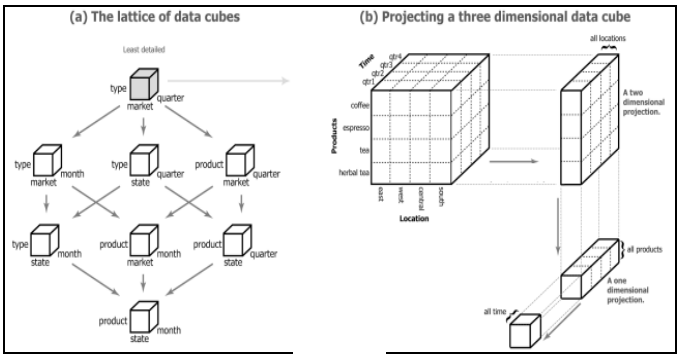

Figure 2

Figure 2 The lattice of data cubes for a data base with three dimensions: Products (with levels Type and Product), Time (with levels Quarter and Month), and Location (with levels Market and State). (b) Several projections of the least detailed data cube in the lattice.

In array based computation, it takes base cuboid X of an n-dimensional data cube stored in an array. That array may be in compressed format if sparse array techniques are used. Then that array may be chunked if too large in hierarchy and the base cuboid itself may need to be computed separately from raw hierarchical data. Its result is in the form of all cube, i.e., all cuboids except the base cuboid. The computed cuboids are stored on disk also as arrays can be used for the data centralization and generalization. It may also be stored together with base cuboid as extended array.

4. m-Dimensional Array

A data cube built from *m* attributes can be stored as an *m*-dimensional array. Each element of the array contains the measure value, such as count. The array itself can be represented as a 1-dimensional array. For example, a 2-dimensional array of size (x x y) can be stored as a 1-dimensional array of size $x*y$, where element (i,j) in the 2-D array is stored in location ($y*i+j$) in the 1-D array. The disadvantage of storing the cube directly as an array is that most data cubes are sparse, so the array will contain many empty elements (zero values).

5. List of Ordered Sets

To save storage space we can store the cube as a sparse array or a list of ordered sets. If we store all cells in the data cube then the resulting data cube will contain(x x y x z) combinations, which is might be large number in combinations. An ordered set representation of the data cube is that in which each attribute value combination is paired with its corresponding count. This representation can be easily stored in a database table to facilitate queries on the data cube.

To compute all 2^n cuboids of an n-dimensional data cube, it may have each cuboids may be computed as a group-by arrangement such as a group like ABC a combination of data hierarchy in which group by A, B, C respectively every member, AB a group by A, B and A group by A alone and last All an empty group-by arrangement.
Sometimes we use cuboids and group-bys interchangeably, but a cuboids may correspond to multiple group-bys arrangement such as AB may be computed from "group by A, B" or "group by B, A" it means the different orders of data element from hierarchy computation may have very different costs and requirements.
Data Cube operators generalize the histogram, cross tabulation, roll-up, drill-down and sub-total constructs required by databases. The following operations can be defined on the data cube.

a. Pivoting involves rotating the cube to change the dimensional orientation of a report or page on display. It may consist of swapping the two dimensions (row and column in a 2D-cube) or introducing another dimension instead of some dimension already present in the cube.
b. Slicing-dicing involves selecting some subset of the cube. For a fixed attribute value in a given dimension, it reports all the values for all the other dimensions. It can be visualized as slice of the data in a 3D-cube. Some dimensions have a hierarchy defied on them. Aggregations can be done at different levels of hierarchy.
c. Rollup or summarization of the data cube can be done by traversing upwards through a concept hierarchy. A concept hierarchy maps a set of low level concepts to higher level, more general concepts. It can be used to summarize information in the data cube. As the values are combined, cardinalities shrink and the cube gets smaller. Generalizing can be thought of as computing some of the summary total cells that contain ANYs, and storing those in favor of the original cells. To reduce the size of the data cube, we can summarize the data by computing the cube at a higher level in the concept hierarchy. A non-summarized cube would be computed at the lowest level. The final result of summarizing the data by introducing the dimensions till the process could be continued to further generalize.
d. Drill-down is similar to Rollup, but is in reverse. A drill-down goes from less detailed data to more detailed data. To drill-down, we can either traverse down a concept hierarchy or add another dimension to the data cube. This is a reversal of the summarization process.

In the multidimensional data hierarchical model for data, there exist in the form of star schema, snowflake schema, or fact constellation schema to work on OLAP cubes or data cubes.

a. Distributive: If the result derived by applying the function to n aggregate values is the same as that derived by applying the function on all the data without partitioning. Functions like count(), sum(), min(), max().
b. Algebraic: Use distributive aggregate functions. If it can be computed by an algebraic function with M arguments (where M is a bounded integer), each of which is obtained by applying a distributive aggregate function. Functions like avg(), min_N(), standard deviation().
c. Holistic: If there is no constant bound on the storage size needed to describe a sub aggregate. Functions like median(), mode(), rank().

A data cube function is a numerical function that can be evaluated at each point in the data hierarchy

6. Key Points

The data cube operation can be expressed in high-level languages like SQL and other query structured languages in the form of a disjunction of group by queries. A query optimizer would then (ideally) combine all the queries into a single query result. Intuitively, this naive algorithm divides the full cubing task into a set of aggregation tasks, one for each cube group, and distributes them for computation using the language framework.
However, as the scale of data increases, it moves towards key challenges that cause the computations to perform poorly and eventually fail. Those are i. size of intermediate data, ii. Size of large groups as described next.

i. Size of Intermediate Data: The first challenge arises from the large size of intermediate data being generated from the various phase of mining process.
ii. Size of Large Groups: The second situation arises from cube groups belonging to cube regions at the bottom part of the cube hierarchy. The reducer that is assigned the latter group essentially has to compute the measure for the entire dataset, which is usually large enough to cause the reducer to take significantly longer time to finish than others or even fail. As the size of the data increases, the number of such groups also increases. We call such groups' reducer unfriendly. A cube region with a significant percentage of reducer-unfriendly groups is called reducer-unfriendly region.

For algebraic measures, this situations can addressed by not processing those groups directly: we can first compute only for those smaller, reducer friendly, groups, then combine those measures to produce the measure for the larger, reducer-unfriendly groups. Such computations are also responsive to suggestion which further decreases the load on the shuffle and reduce phases. For holistic measures, however, computations for larger groups cannot be assembled from their smaller groups, and hence any one may need a different approach for computing the data cubes.

7. Conclusion

Data Mining Process is fast gaining importance for business data analysis using large amounts of data now available in data warehouses. Data Cubes aggregations are an important function of OLAP queries and can benefit. Multidimensional databases model the multi-dimensionality of data intuitively, providing support for complex analytical queries, also being responsive to requirement. Summary results in the data cube can be used to perform data mining through attribute focusing methods. I have presented a general discussion regarding data cube computation with the help of data mining to perform attribute focusing on the data cube. In this article, I presented techniques for computing the attributes of multidimensional data cubes with data mining.

8. References

1. Sarawagi S., Agrawal R., and Gupta A., "On Computing the Data Cube", Research Report 10026, IBM Almaden Research Center, San Jose, California, 1996
2. Antoaneta Ivanova, Boris Rachev , Multidimensional models - Constructing DATA CUBE : International Conference on Computer Systems and Technologies – CompSysTech, 2004.
3. Venky Harinarayan, Anand Rajaraman, Jeffrey D. U1lman, K. Beyer and R. Ramakrishnan. Implementing Data Cubes Efficiently,SIGMOD, Montreal, Canada,1996
4. Bottom-up computation of sparse and iceberg CUBEs. In SIGMOD 1999.
5. IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, VOL. 9, NO. 2, APRIL-JUNE 2003.
6. Arnab Nandi, Cong Yu, Philip Bohannon, and Raghu Ramakrishnan. Data Cube Materialization and Mining over MapReduce, Transactions On Knowledge And Data Engineering, 2012.
7. B.-C. Chen, L. Chen, Y. Lin, and R. Ramakrishnan. Prediction cubes. International Conference on Very Large Data Bases (VLDB'05), Trondheim, Norway, Aug. 2005.
8. S. Chaudhuri and U. Dayal. An overview of data warehousing and OLAP technology. SIGMOD, 1997.
9. R. Fagin, R. V. Guha, R. Kumar, J. Novak, D. Sivakumar, and A. Tomkins. Multi-structural databases. In Process ACM SIGMOD-SIGACT-SIGAR, 2005.
10. Principles of Database Systems (PODS'05) Baltimore, MD, 2005.
11. Molina.H, J. Ullman, J. Windom, DataBase Systems: The Complete Book, Prentice Hall, 2002
12. Codd E. F., "Providing OLAP to user-analysts : An IT mandate", Technical Report, E.F. Codd and Associates, 1993.
13. http://cis.ieee.org
14. http://www2.cs.uregina.ca
15. http://cs.nju.edu.cn
16. http://www.aihw.gov.au/procedures-data-cubes
17. http://www.computerworld.com/s/article/91640/Data_Cubes.