International Journal of Engineering Applied Sciences and Technology, 2018
Vol. 3, Issue 8, ISSN No. 2455-2143, Pages 66-69
Published Online December 2018 in IJEAST (http://www.ijeast.com)

HORIZONTAL DATA MINING GENERATOR USING CUBE FOR BUSINESS INTELLIGENCE

Anaswara Venunadh
Department of ISE
MVJ college of Engineering, Bangalore, Karnataka, India

Abstract— **Business Intelligence (BI) systems are being increasingly used by organizations and considered as an advantage, which goal is to offer access to information in a timely manner to support the decision-making process. However it should be noted that this is an area of activity with peculiar characteristics that must be taken into account. Here a simple methods to generate SQL code to return aggregated columns in a horizontal tabular layout, returning a set of numbers instead of one number per row. Horizontal aggregations build data sets with a horizontal layout (e.g. point-dimension, observation-variable, instance-feature), which is the standard layout required by most data mining algorithms. We propose three fundamental methods to evaluate horizontal aggregations: CASE, SPJ and PIVOT. Also performance evaluations of three methods are calculated. Here data from database is used for constructing a cube and then horizontally aggregating them, to get the required data set. This paper will give an idea about how horizontal aggregation can be used for business intelligence.**

Keywords— **Horizontal aggregation, cube, data Mining, CASE, PIVOT, SPJ and Business intelligence.**

I. INTRODUCTION

In today's highly competitive business scenario, it is imperative to continuously stay updated on technology. Almost every business works hard to achieve success, but what makes one stand out is the understanding which technology to use at what time so as to render maximum gains; gains not only in terms of money, but in terms of driving a change such that numerous lives are befitted. In a relational database, especially with normalized tables, a significant effort is required to prepare a summary data set that can be used as input for a data mining or statistical algorithm .Most algorithms require as input a data set with a horizontal layout, with several records and one variable or dimension per column. That is the case with models like clustering, classification, regression and PCA; consult .Each research discipline uses different terminology to describe the data set. In data mining the common terms are point-dimension. Statistics literature generally uses observation-variable as given in article given by Fayyad et al.(1996)[8]. Machine learning research uses instance-feature and the part of neural networks given by K. Amarendra et al.(2009)[2].

Organizations today are collecting data at every level of their business and in volumes that in the past were unimaginable image of owners' license information and to track illegal copies.Data sets are stored in different database systems or in files with distinctive formats, all reflecting business process, application, program software, or information type dependencies.

This article introduces a class of aggregate functions that can be used to build data sets in a horizontal layout (denormalized with aggregations), automating SQL query writing and extending SQL capabilities. Here shows evaluating horizontal aggregations are a challenging and interesting problem and introduced alternative methods and optimization for their efficient evaluation. LAP tools generate SQL code to transpose results (sometimes called PIVOT) as described in C. Cunningham et al.(2004)[15],. Transposition can be more efficient if there are mechanisms combining aggregation and transposition together. With such limitations in mind, propose a new class of aggregate functions that aggregate numeric expressions and transpose results to produce a data set with a horizontal layout. Functions belonging to this class are called horizontal aggregations.

Horizontal aggregations represent an Extended form of traditional SQL aggregations, which return a set of values in a horizontal layout (somewhat similar to a multidimensional vector), instead of a single value per row explained by Carlos Ordonez et al.(2011)[1],. This article explains how to evaluate and optimize horizontal aggregations generating standard SQL code.

The rest of the paper is organized as follows. Existing system and proposed system are explained in section II and III. Proposed solution along with algorithm for implementation in section IV and V. Concluding remarks are given in section VI

II. EXISTING SYSTEM

Although business intelligence does not tell business users what to do or what will happen if they take a certain course, neither is BI only about generating reports. Rather, BI offers a way for people to examine data to understand trends

International Journal of Engineering Applied Sciences and Technology, 2018
Vol. 3, Issue 8, ISSN No. 2455-2143, Pages 66-69
Published Online December 2018 in IJEAST (http://www.ijeast.com)

and derive insights. Existing SQL (J. Clear et al. (1999) [3]) aggregations have limitations to prepare data sets because they return one column per aggregated group. It cannot handle complicated relationship between features. Data set for analysis is generally the most time consuming task in a data mining project, requiring many complex SQL queries, joining tables and aggregating columns. No system focuses in the Data Mining area with involvement of Knowledge Cubes. Performance evaluation on the common data with different scenarios was not evaluated properly in the existing system.

Business intelligence is the practice of taking large amounts of corporate data and turning it to usable information is given in Rouhani, S et al.(2016)[13]. This practice enables companies to derive analysis that can be used to make profitable actions. The process of converting corporate data to usable information is time consuming, and involves various factors such as data models, data sources, data warehouses and business models, among others.

Setting up a successful business intelligence environment involves having the right tools and systems in place. It requires having business analysts and owners who can guide the initiative. There are various factors to take into account when setting up a business intelligence environment, including the data types to be analyzed, the right tools for the job and determining how the data will be integrated for business intelligence analysis.

Disadvantages of Existing System:

- Sequential queries are not possible in this system.
- Faster retrieval is not possible.
- Time required for the data set preparation is comparatively high.
- One column per aggregation only achieved in this system

III. PROPOSED SYSTEM

Building a suitable data set for business intelligence purposes is a time-consuming task. This task generally requires writing long SQL statements or customizing SQL code if it is automatically generated by some tool. There are two main ingredients in such SQL code: joins and aggregations; we focus on the second one. The most widely-known aggregation is the sum of a column over groups of rows. Some other aggregations return the average, maximum, minimum or row count over groups of rows. There exist many aggregation functions and operators in SQL. Unfortunately, all these aggregations have limitations to build data sets for data mining purposes.

The main reason is that, in general, data sets that are stored in a relational database (or a data warehouse) come from On-Line Transaction Processing (OLTP) systems where database schema is highly normalized. But data mining, statistical or machine learning algorithms generally require aggregated data in summarized form. Based on current available functions and clauses in SQL, a significant effort is required to compute aggregations when they are desired in a cross tabular (horizontal) form, suitable to be used by a data mining algorithm. Such effort is due to the amount and complexity of SQL code that needs to be written, optimized and tested. There are further practical reasons to return aggregation results in a horizontal (cross-tabular) layout. Standard aggregations are hard to interpret when there are many result rows, especially when grouping attributes have high cardinalities. The ten main challenges in collecting data in business intelligence are as follows:

1. Unstructured Data of Business Intelligence
2. Delivering self-service reporting/ analysis
3. Reporting/ analyzing across multiple systems
4. Unlocking data buried in systems
5. Reducing the cost of producing reports
6. Absence of Execution and Training
7. Scaling Up for High Dimensional Data and High Speed Streams.
8. Sequential and Time Series Data.
9. Mining Complex Knowledge from Complex Data.
10. Data Mining in a Network Setting.

Today, the most successful companies are those that can respond quickly and flexibly to market changes and opportunities (i.e., they are agile). The key to this response is the effective and efficient use of data and information (Jourdan, Z et al. (2008) [14]). Data mining is the process of mining knowledge or data from a large repositories or data warehouses. So a suitable way should be there for preparing the data set for analysis. The data set preparation should be done in such a way that easy retrieval should be possible and data retrieved should be accurate. The various stages in business intelligence are

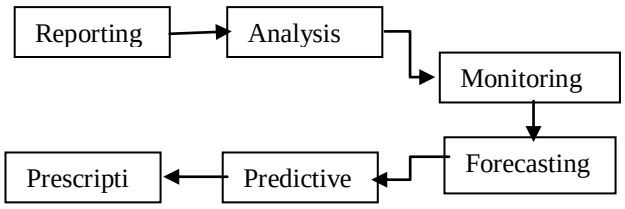

Fig 1. Stages in Business Intelligence

The methods which are using now for data set preparation in data mining are

- Statistical data mining
- Tree based methods
- Neural networks
- Near neighbor method

While considering all the above methods, each having problem regarding easy and accurate retrieval of data. The traditional methods of statistics and pattern recognition are

International Journal of Engineering Applied Sciences and Technology, 2018
Vol. 3, Issue 8, ISSN No. 2455-2143, Pages 66-69
Published Online December 2018 in IJEAST (http://www.ijeast.com)

either parametric based on a family of models with a small number of parameters, or non parametric in which the models used are totally flexible. One of the impacts of neural network methods on pattern recognition has been to emphasize the need in large-scale practical problems for something in between, families of models with large but not unlimited flexibility given by a large number of parameters. The two most widely used neural network architectures, multi-layer perceptions and radial basis functions (RBFs), provide two such families (and several others already existed in statistics). Another difference in emphasis is on 'on-line' methods, in which the data are not stored except through the changes the learning algorithm has made. The theory of such algorithms is studied for a very long stream of examples, but the practical distinction is less clear, as this stream is made up either by repeatedly cycling through the training set or by sampling the training examples (with replacement). In contrast, methods which use all the examples together are called 'batch' methods. It is often forgotten that there are intermediate positions, such as using small batches chosen from the training set.

VI. PROPOSED ALGORITHM

A.**Pivoted generalized & horizontally suppressed algorithm:**

In Generalized algorithm, we are trying to disclose the generalized data like we are not going to disclose exact data in the column. A generalized aggregated data will be provided. In addition, the normal exact data were suppressed to provide only the aggregated values

STEPS:

INPUT: Private table PT :quasi-identifier QI=(A1,……,An),disjoint subsets of QI known as identifying,More,and Most where QI=identifying∪More∪Most,k constraint; domain generalization hierarchies DGHaj,where i=1….n.
OUTPUT: MT containing a generalization of PT [QI]
ASSUMES|PT|≥k
METHOD:

- Freq← a frequency list containing distinct sequences of values of PT [QI], along with the number of occurrences of each sequence.
- Generalize each Ai€QI in freq until its assigned values satisfy k
- Test 2- and 3- combinations of identifying, more and most and let outliers store those cell combinations not having k occurrences.
- Data holder decides whether to generalize an Aj € QI based on outliers and if so identifies the Aj to generalize.freq contains the generalized result.
- Repeat steps 3 and 4 until the data holder no longer elects to generalize.
- Automatically suppress a value having a combination in outliers, where precedence is given to the value occurring in the most number of combinations of outliers.

V. IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system.
Modules Description:

1. Authenticate Module
2. Case State Module
3. SPJ Check Module
4. PIVOT check module
5. Knowledge Cube Construction Module
6. MDX Query Check Module
7. Performance Comparison & Evaluation Module

Authenticate Module describes the interface implemented by authentication technology providers. Login Modules are plugged in under applications to provide a particular type of authentication. The case statement returns a value selected from a set of values based on Boolean expressions (C. Galindo et al. (1997) [6]). CASE can be used in any statement or clause that allows a valid expression. The basic idea is to create one table with a vertical aggregation for each result column, and then join all those tables to produce another table. It is based on standard relational algebra operators (SPJ queries).It is necessary to introduce an additional table F0 that will be outer joined with projected tables to get a complete result set. The PIVOT method internally needs to determine how many columns are needed to store the transposed table and it can be combined with the GROUP BY clause. The basic syntax to exploit the PIVOT operator is to compute a horizontal aggregation assuming one by column from the right key columns. In knowledge cube construction module there is three main configurations:

1. ConFigure Data source
2. ConFigure Dimensions
3. ConFigure Cubes.

By using Multidimensional query we are going to provide a data security like the owner of that data can view exact data and the other users can view only a partial data. Here we can access Cube instead of table. In performance comparison and evaluation module, we are going to compare the performance of SPJ, CASE and Pivot method and going to find the efficiency of each and every method.

VI. CONCLUSION

An effective BI system supports the needs of a variety of users — executives, managers, analysts, power users and casual business users — in an integrated and comprehensive manner, Cognos (2008) [7]. Executives require information that is highly summarized and directly relevant to their key initiatives. In this project, described the class of extended aggregate functions, called horizontal aggregations which help preparing data sets for business

intelligence and OLAP cube exploration is explained in this project. Basically, a horizontal aggregation returns a set of numbers instead of a single number for each group, resembling a multi-dimensional vector. Consider various factors like scalability of number of features and instances, automation for handling large, heterogeneous data.

Horizontal aggregations produce tables with fewer rows, but with more columns. Thus query optimization techniques used for standard (vertical) aggregations (C. Ordonez,(2004)[16]) are inappropriate for horizontal aggregations. Plan to develop more complete I/O cost models for cost-based query optimization. Want to study optimization of horizontal aggregations processed in parallel in a shared-nothing DBMS architecture. Cube properties can be generalized to multi-valued aggregation results produced by a horizontal aggregation. Need to understand if horizontal aggregations can be applied to holistic functions (e.g. rank ()). Optimizing a workload of horizontal aggregation queries is another challenging problem.

VIII. REFERENCE

[1] Carlos Ordonez and Zhibo Chen,(2011),' Horizontal Aggregations in SQL to Prepare Data Sets for Data Mining Analysis ',ieee transactions on knowledge and data engineering, vol. 24, no. 4.

[2] K. Amarendra, K.V. Lakshmi & K.V. Ramani (2009)'Research Of Data Mining Based On Neural Networks', Special Issue Of International Journal Of Computer Science & informatics (ijcsi), issn (print) : 2231–5292, vol.- ii, issue-1, 2.

[3] J. Clear, D. Dunn, B. Harvey, M.L. Heytens, and P. Lohman (1999).' Non-stop SQL/MX primitives for knowledge discovery'. In ACM KDD Conference, pages 425–429.

[4] C. Galindo-Legaria and A. Rosentahl,(1997)"Outer Join Simplification and Reordering for Query Optimization," ACM Trans. Database Systems, vol.22, no.1, pp.43-73,

[5] E.F. Codd.(1997) 'Extending the database relational model to capture more meaning'. ACM TODS, 4(4):397–434.

[6] C. Galindo-Legaria and A. Rosenthal (1997).' Outer join simplification and reordering for query optimization'. ACM TODS, 22(1):43–73.

[7] Cognos, (2008), "BI for Business Users," January (white paper available at www.cognos.com).

[8] Fayyad, Usama, Gregory Piatetsky-Shapiro, Padhraic Smyth,(1996) From Data Mining to Knowledge Discovery in Databases, .

[9] Jiawei Han, Micheline Kamber,(2011) Data Mining: Concepts and Techniques, London: Academic Press, 5.

[10] X. Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q. Yang, H. motoda, G.J. MClachlan, A. Ng, B. Liu, P.S. Yu, Z. Zhou, M. Steinbach, D. J. Hand, D. Steinberg, (2011) Top 10 Algorithms in Data Mining, Knowl Inf Syst 141-37.

[11] Hongjian Qiu, Yihua Huang, Rong Gu, Chunfeng Yuan, (2014)"YAFIM: A Parallel Frequent Itemset Mining Algorithm with Spark", IEEE 28th International Parallel & Distributed Processing Symposium Workshops.

[12] Zhao, Y., Deshpande, P. M. & Naughton, J. F. (1997), An array-based algorithm for simultaneous multidimensional aggregates, in `Proc. 1997 ACM-SIGMOD Int. Conf. Management of Data', Tucson, Arizona, pp. 159{170.

[13] Rouhani, S., Ashrafi, A., Zare, A., Afshari, S., Irani, Z., Irani, Z.(2016): The impact model of business intelligence on decision support and organizational benefits. Journal of Enterprise Information Management 29, 19-50.

[14] Jourdan, Z., Rainer, R.K., Marshall, T.E.(2008): Business Intelligence: An Analysis of the Literature. Information Systems Management 25, 121-1.

[15] C. Cunningham, G. Graefe, and C.A. Galindo-Legeria(2004), "PIVOTAND UNPIVOT: Optimization and Execution Strategies in an RDBMS," Proc: 13 th Int'l Conf. Very Large Data Bases (VLDS'04), pp.998-1009.

[16] C. Ordonez,(2004) "Vertical and Horizontal Percentage Aggregations," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD'04),pp.866-871.