ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 3, March 2014

A Fast Distributed Mining of Association Rules In Horizontally Distributed Database

Rajkumar.S [1], V.Elavarasi[2]

M.Tech (CSE) Student, Department of CSE, S.R.M. University,Ramapuram, Chennai.India[1]

Assistant Professor, Department of CSE, S.R.M. University,Ramapuram, Chennai,India[2]

Abstract: Data mining can extract important knowledge from large data collections, but sometimes these collections are split among various parties. This paper addresses a fast distributed mining of association rules over horizontally distributed data. While preparing a data set for analysis is generally the most time consuming task in a data mining,requiring numerous complex SQL queries, joining tables and aggregating columns. Existing SQL aggregations have limitations to prepare data sets because they return one column per aggregated group. In general, a significant manual effort is required to build data sets, where a horizontal layout is required. The proposed is simple, yet powerful, 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. This new class of functions is called horizontal aggregations. Horizontal aggregations build data sets with a horizontal de normalized layout (e.g. point-dimension, observation-variable, instance-feature), which is the standard layout required by most data mining algorithms. The proposed method used three categories to evaluate horizontal aggregations: CASE: Exploiting the programming CASE construct; SPJ: Based on standard relational algebra operators (SPJ queries); PIVOT: Using the PIVOT operator, which is offered by some DBMSs. Experiments with large tables compare the proposed query evaluation methods. A CASE method has similar speed to the PIVOT operator and it is much faster than the SPJ method. In general, the CASE and PIVOT methods exhibit linear scalability, whereas the SPJ method does not.

Keywords: SPJ Queries, PIVOT, SQL Aggregations, CASE Method, Horizontal Aggregation.

I. INTRODUCTION

Data mining methodology has emerged as a means of identifying patterns and trends from large quantities of data. Data mining go hand in hand: most tools operate by gathering all data into a central site, then running an algorithm against that data.. This paper addresses the problem of computing association rules within such a scenario. We assume homogeneous databases: All sites have the same schema, but each site has information on different entities. The goal is to produce association rules that hold globally, while limiting the information shared about each site. Computing association rules without disclosing individual transactions is straight forward. 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. Machine learning research uses instance-feature. This paper introduces a new class of aggregate functions that can be used to build data sets in a horizontal layout (de normalized with aggregations), automating SQL query writing and extending SQL capabilities.

We show evaluating horizontal aggregations is a challenging and interesting problem and we introduced alternative methods and optimizations for their efficient evaluation.

II. LITERATURE SURVEY

We have to investigation the Data Mining Literature Survey: Data mining originate its name from the similarities between searching for valuable business information in a large database —for example, finding linked products in

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 3, March 2014

gigabytes of store scanner data —and mining a mountain for a vein of valuable ore. Both processes require either sifting through an immense amount of material, or intelligently probing it to find exactly where the value resides. Given databases of sufficient size and quality, data mining technology can generate new business opportunities by providing these capabilities:

Automated prediction of trends and behaviors: Data mining automates the process of finding predictive information in large databases. Questions that traditionally required extensive hands-on analysis can now be answered directly from the data quickly. A typical example of a predictive problem is targeted marketing. Data mining uses data on past promotional mailings to identify the targets most likely to maximize return on investment in future mailings. Other predictive problems include forecasting impoverishment and other forms of default, and identifying segments of a population likely to respond similarly to given events.

Automated discovery of previously unknown patterns: Data mining tools sweep through databases and identify previously hidden patterns in one step. An example of pattern discovery is the analysis of retail sales data to identify seemingly unrelated products that are often purchased together. Other pattern discovery problems include detecting fraudulent credit card transactions and identifying anomalous data that could represent data entry keying errors.

SQL extensions are defining aggregate functions for association rule mining. Their optimizations have the purpose of avoiding the joins to convey unit (cell) formulas, but are not optimized to perform partial Transposition for each group of result rows. Conor Cunningalam proposed an optimization and Execution strategies in an RDBMS which uses two operators i.e ., PIVOT operator on tabular data that exchange rows and columns enabled data transformations are useful in data modeling, data analysis, and data presentation. They can quite easily be implemented inside a query processor system, much like select, project, and join operator. Such design provides the opportunities for better performance, both during query optimization and query execution. Pivot is an extension of Group By with an unique restrictions and optimization opportunities, and this makes it is very simple to introduce incrementally on top of existing grouping implementations. H Wang.C.Zaniolo proposed a s mall but Complete SQL Extension for Data Streams Data Mining and. This technique is a powerful database language and system that enables users develop complete data-intensive applications in SQL by writing new aggregates and table functions in SQL, rather than in procedural languages as in current Object -Relational systems .

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy n company strength. Once these things r satisfied, ten next steps is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external suppor. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration r taken into account for developing the proposed system. As horizontal aggregations are capable of producing data sets that can be used for real world data mining activities

III. RELATED WORK

Association Rule mining is one of the most important data mining tools used in many real life applications. It is used to reveal unexpected relationships in the data. In this paper, we will discuss the problem of computing association rules within a horizontally partitioned database. We assume homogeneous databases. All sites have the same schema, but each site has information on different entities. The goal is to produce associate ion rules that hold globally, while limiting the information shared about each site to preserve the privacy of data in each site.

Association Rule Mining: Association rule mining finds interesting associations and/or correlation relationships among large sets of data items. Association rules show attributes value conditions that occur frequently together in a given dataset.

Apriori Algorithm : The Apriori Algorithm proposed to finds frequent items in a given data set using the ant monotone constraint. Apriori is an influential algorithm in market basket analysis for mining frequent item sets for Boolean association rules. The name of Apriori is based on the fact that the algorithm uses a prior knowledge of frequent itemset properties. Apriori employs an iterative approach known as a level wise search, where k item sets are

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 3, March 2014

used to explore (k+1) itemsets .Apriori algorithm is an in fluential algorithm for mining frequent itemsets for Boolean association rules. This algorithm contains a number of passes over the database. During pass k, the algorithm finds the set of frequent itemsets Lk of length k that satisfy the minimum support requirement. Apriori is designed to operate on databases containing transactions. The purpose of the Apriori Algorithm is to find associations between different sets of data. It is sometimes referred to as "Market Basket Analysis". Each set of data has a number of items and is called a transaction. The output of Apriori is sets of rules that tell us how often items are contained in sets of data. Verification if the auditor is convinced with the data integrity; the auditor erases the local data.

IV . OBJECTIVES AND MOTIVATIONS

Objectives Generally, data mining (sometimes called data or knowledge discovery database (KDD) is the process of analyzing data from different perspectives and summarizing it into useful information. Information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among different fields in large relational databases. Building a suitable data set for data mining 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 aggregations 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 schemas are 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. To perform analysis of exported tables into spreadsheets it may be more convenient to have aggregations on the same group in one row (e.g. to produce graphs or to compare data sets with repetitive information). OLAP tools generate SQL code to transpose results (sometimes called PIVOT). Transposition can be more efficient if there are mechanisms combining aggregation and transposition together. With such limitations in mind, we 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. This article explains how to evaluate and optimize horizontal aggregations generating standard SQL code.

V. DATA MINING TECHNIQUES

The most commonly used techniques in data mining are:

- **Clustering**: Data items are grouped according to logical relationships or consumer preferences. For example, data can be mined to identify market segments or consumer affinities.
- **Associations Rule**: Data can be mined to identify associations. The beer-diaper example is an example of associative mining.
- **Sequential patterns**: Data is mined to anticipate behavior patterns and trends. For example, an outdoor equipment retailer could predict the likelihood of a backpack being purchased based on a consumer's purchase of sleeping bags and hiking shoes.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 3, March 2014

- **Artificial neural networks**: Non-linear predictive models that learn through training and resemble biological neural networks in structure.

- **Genetic algorithms**: Optimization techniques that use processes such as genetic combination, mutation, and natural selection in a design based on the concepts of natural evolution.

- **Decision trees**: Tree-shaped structures that represent sets of decisions. These decisions generate rules for the classification of a dataset. Specific decision tree methods include Classification and Regression Trees (CART) and Chi Square Automatic Interaction Detection (CHAID) . CART and CHAID are decision tree techniques used for classification of a dataset. They provide a set of rules that you can apply to a new (unclassified) dataset to predict which records will have a given outcome.

- **Nearest neighbor method**: A technique that classifies each record in a dataset based on a combination of the classes of the *k* record(s) most similar to it in a historical dataset (where *k* 1). Sometimes called the *k*-nearest neighbor technique.

- **Rule induction**: The extraction of useful if-then rules from data based on statistical significance.

- **Data visualization**: The visual interpretation of complex relationships in multidimensional data. Graphics tools are used to illustrate data relationships.

VI. HORIZONTAL AGGREGATIONS USED IN SQL

Introduce a new class of aggregations that have similar behavior to SQL standard aggregations, but which produce tables with a horizontal layout. In contrast, we call standard SQL aggregations vertical aggregations since they produce tables with a vertical layout. Horizontal aggregations just require a small syntax extension to aggregate functions called in a SELECT statement. Alternatively, horizontal aggregations can be used to generate SQL code from a data mining tool to build data sets for data mining analysis. We start by explaining how to automatically generate SQL code

1 .SQL Code Generation: The main goal is to define a template to generate SQL code combining aggregation and transposition (pivoting). A second goal is to extend the SELECT statement with a clause that combines transposition with aggregation. Consider the following GROUP BY query in standard SQL that takes a subset L1 ... Lm from D 1 ,... D p

```
SELECT L1 ,... Lm , sum(A)
FROM F
GROUP BY L1 ... Lm.
```

2. Proposed Syntax in Extended SQL : We now turn our attention to a small syntax extension to the SELECT statement, which allows understanding our proposal in an intuitive manner. We must point out the proposed extension represents non -standard SQL because the columns in the output table are not known when the query is parsed.

3. SQL Code Generation: Query Evaluation Methods We proposes three methods to evaluate horizontal aggregations. The first method relies only on relational operations. That is, only doing select, project, join and aggregation queries; we call it the SPJ method. The second form relies on the SQL "case" constructs; we call it the CASE method. Each table has an index on its primary key for efficient join processing.. The third method uses the built in PIVOT operator, which transforms rows to columns (e.g. transposing). An overview of the main steps to be explained below (for a sum ()) aggregation.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 3, March 2014

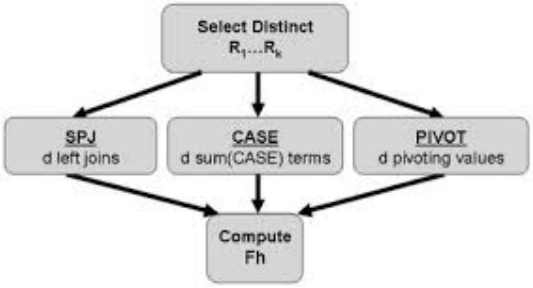

Fig1.SPJ Method

There are three method used as follows:

1 .SPJ Method: The SPJ method is interesting from a theoretical point of view because it is based on relational operators only. The basic idea is to create one table with a vertical aggregation for each result column, and then join all those tables to produce FH.

2. CASE Method: This method uses the case programming construct available in SQL. The case statement returns a value selected from a set of values based on boolean expressions. From a relational database theory point of view this is equivalent to doing a simple projection/aggregation query where each non – key value is given by a function that returns a number based on some conjunction of conditions.

3. PIVOT Method: The PIVOT Method used PIVOT operator which is a built in operator in a commercial DBMS. Since this operator can perform transposition it can help evaluating horizontal aggregations. 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.

VII .PROPOSED SYSTEM

The proposed system is horizontal aggregations provide several unique features and advantages. First, they represent a template to generate SQL code from a data mining tool. Such SQL code automates writing SQL queries, optimizing them and testing them for correctness. This SQL code reduces manual work in the data preparation phase in a data mining project. Second, since SQL code is automatically generated it is likely to be more efficient than SQL code written by an end user. For instance, a person who does not know SQL well or someone who is not familiar with the database schema (e.g. a data mining practitioner). Therefore, data sets can be created in less time. Third, the data set can be created entirely inside the DBMS. In modern database environments it is common to export de normalized data sets to be further cleaned and transformed outside a DBMS in external tools (e.g. statistical packages). Unfortunately, exporting large tables outside a DBMS is slow, creates inconsistent copies of the same data and compromises database security. Therefore, we provide a more efficient, better integrated and more secure solution compared to external data mining tools. Horizontal aggregations just require a small syntax extension to aggregate functions called in a SELECT statement. Alternatively, horizontal aggregations can be used to generate SQL code from a data mining tool to build data sets for data mining analysis.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 3, March 2014

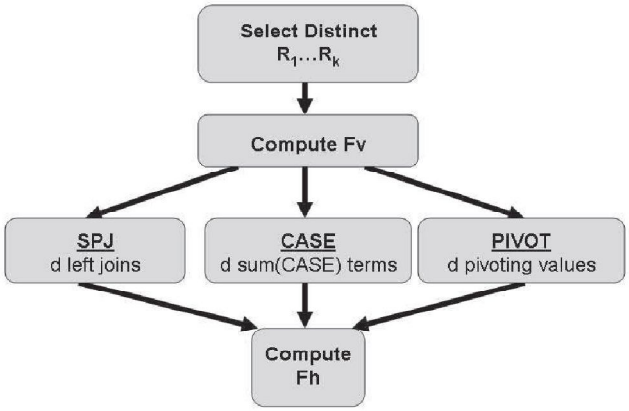

Figure.2.Proposed System Process Flow

VIII. CONCLUSION

We introduced a new class of extended aggregate functions, called horizontal aggregations which help preparing data sets for data mining and OLAP cube exploration. Specifically, horizontal aggregations are useful to create data sets with a horizontal layout, as commonly required by data mining algorithms and OLAP cross-tabulation. Basically, a horizontal aggregation returns a set of numbers instead of a single number for each group, resembling a multi-dimensional vector. A minimal, extension to SQL standard aggregate functions to compute horizontal aggregations which just requires specifying sub grouping columns inside the aggregation function call. From a query optimization perspective, we proposed three query evaluation methods. The first one (SPJ) relies on standard relational operators. The second one (CASE) relies on the SQL CASE construct. The third (PIVOT) uses a built-in operator in a commercial DBMS that is not widely available. The SPJ method is important from a theoretical point of view because it is based on select, project and join (SPJ) queries. The CASE method is our most important contribution. It is in general the most efficient evaluation method and it has wide applicability since it can be programmed combining GROUP-BY and CASE statements. We proved the three methods produce the same result.

To evaluate horizontal aggregations using standard SQL without either joins or ”case” constructs using standard SQL operators. Our proposed horizontal aggregations can be used as a database method to automatically generate efficient SQL queries with three sets of parameters: grouping columns, subgrouping columns and aggregated column. The fact that the output horizontal columns are not available when the query is parsed (when the query plan is explored and chosen) makes its evaluation through standard SQL mechanisms infeasible. Our experiments with large tables show our proposed horizontal aggregations evaluated with the CASE method have similar performance to the built-in PIVOT operator. We believe this is remarkable since our proposal is based on generating SQL code and not on internally modifying the query optimizer. Both CASE and PIVOT evaluation methods are significantly faster than the SPJ method. Pre computing a cube on selected dimensions produced acceleration on all methods.

REFERENCES

1.R. Agrawal and R. Srikant. Fast algorithms for mining association rulesin large databases. In VLDB, pages 487–499, 1994.
2.R. Agrawal and R. Srikant. Privacy-preserving data mining. In SIGMODConference, pages 439–450, 2000.
3.D. Beaver, S. Micali, and P. Rogaway. The round complexity of secureprotocols. In STOC, pages 503–513, 1990.
4.M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. In Crypto, pages 1–15, 1996.
5.Ben-David, N. Nisan, and B. Pinkas. FairplayMP - A system for secure multi-party computation. In CCS, pages 257–266, 2008.
6.J.C. Benaloh. Secret sharing homomorphisms: Keeping shares of a secretsecret. In Crypto, pages 251–260, 1986.
7. J. Brickell and V. Shmatikov. Privacy-preserving graph algorithms inthe semi-honest model. In ASIACRYPT, pages 236–252, 2005.
8. D.W.L. Cheung, J. Han, V.T.Y. Ng, A.W.C. Fu, and Y. Fu. A fast distributed algorithm for mining association rules. In PDIS, pages 31–42, 1996.
9.D.W.L Cheung, V.T.Y. Ng, A.W.C. Fu, and Y. Fu. Efficient mining of association rules in distributed databases. IEEE Trans. Knowl. Data Eng., 8(6):911–922, 1996.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 3, March 2014

10.T. ElGamal. A public key cryptosystem and a signature scheme based ondiscrete logarithms. IEEE Transactions on Information Theory, 31:469
11.C. Cunningham, G. Graefe, and C.A. Galindo-Legaria. PIVOT and UNPIVOT: Optimization and execution strategies in an RDBMS. In Proc. VLDB Conference.
12.J. Gray, A. Bosworth, A. Layman, and H. Pirahesh. Data cube: A relational aggregation operator generalizing group- by, cross-tab and subtotal. In ICDE Conference.
13.J. Han and M. Kamber. Data Mining: Concepts and Techniques . Morgan Kaufmann, San Francisco, 1st edition 2001.