End-to-End Methodology for Mining from Datasets

Fahmida Begum[1], K.Kishore[2]

[1]M.Tech Student, Dr.K.V.Subba Reddy College of Engineering for Women, Kurnool, AP

[2]Asst.Professor, Dr.K.V.Subba Reddy College of Engineering for Women, Kurnool, AP

Abstract:

Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases. Data mining is widely used domain for extracting trends or patterns from historical data. Generally, data sets that are stored in a relational database or a data warehouse come from On-Line Transaction Processing (OLTP) systems in which database schemas are highly normalized. But data mining, statistical or machine learning algorithms generally require aggregated data in summarized form. Also many data mining algorithms require the result to be transformed into tabular format. Tabular datasets are the suitable input for many data mining approaches. In any competitive business, success is based on the ability to make an item more appealing to customers than the competition. A number of questions arise in the context of this task: *how do we formalize and quantify the competitiveness between two items? Who are the main competitors of a given item? What are the features of an item that most affect its competitiveness?* Despite the impact and relevance of this problem to many domains, only a limited amount of work has been devoted toward an effective solution. In this paper, we present a formal definition of the competitiveness between two items, based on the market segments that they can both cover. Our evaluation of competitiveness utilizes customer reviews, an abundant source of information that is available in a wide range of domains. We present efficient methods for evaluating competitiveness in large review datasets and address the natural problem of finding the top-k competitors of a given item. Finally, we evaluate the quality of our results and the scalability of our approach using multiple datasets from different domains.

Keywords: *Data Mining, Dataset, Vertical Aggregation, Horizontal Aggregation*

1.INTRODUCTION

The term data mining refers to extracting or "mining" knowledge from large amounts of data. Many people treat data mining as a synonym for another popularly used term, Knowledge Discovery from Data, or KDD. Knowledge discovery as a process consists of an iterative sequence of steps consists like Data cleaning, Data integration , Data selection , Data transformation , Data mining, Pattern evaluation and Knowledge presentation. From this we can define data mining as an essential process where intelligent methods are applied in order to extract data patterns. Data mining uses mathematical analysis to derive patterns and trends that exist in data. Typically, these patterns cannot be discovered by traditional data exploration because the relationships are too complex or because there is too much data. These patterns and trends can be collected and defined as a data mining model. Mining models can be applied to specific scenarios, such as Forecasting, Risk and probability, Recommendations, Finding sequences and Grouping.

Based on this view, the architecture of a typical data mining system is shown in the figure 1 have the following major components Database, data warehouse, Worldwide Web, or other information repository: This is one or a set of databases, data warehouses, spreadsheets, or other kinds of information repositories. Data cleaning and data integration techniques may be performed on the data. **Database or data warehouse server**: The database or data warehouse server is responsible for fetching the relevant data, based on the user's data mining request. Knowledge base: This is the domain knowledge that is used to guide the search or evaluate the interestingness of resulting patterns. Such knowledge can include concept hierarchies, used to organize attributes or attribute values into different levels of abstraction.

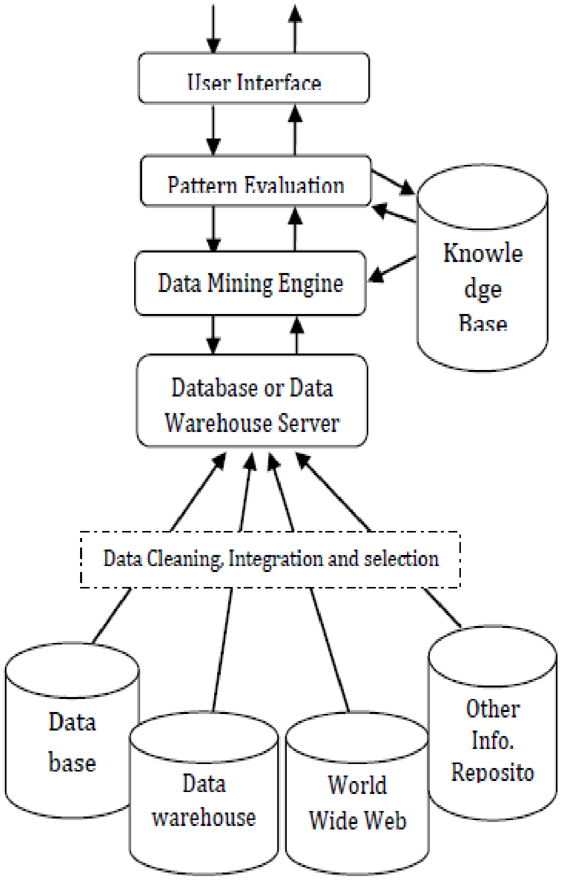

Fig 1. Architecture of a typical Data mining System

Database or data warehouse server: The database or data warehouse server is responsible for fetching the relevant data, based on the user's data mining request. Knowledge base: This is the domain knowledge that is used to guide the search or evaluate the interestingness of resulting patterns. Such knowledge can include concept hierarchies, used to organize attributes or attribute values into different levels of abstraction.

Data mining engine: This is essential to the data mining system and ideally consists of a set of functional modules for tasks such as characterization, association and correlation analysis, classification, prediction, cluster analysis, outlier analysis, and evolution analysis. **Pattern evaluation module**: This component typically employs interestingness measures and interacts with the data mining modules so as to *focus* the search toward interesting patterns. It may use interestingness thresholds to filter out discovered patterns. **User interface**: This module communicates between users and the data mining system, allowing the user to interact with the system by specifying a data mining query or task, providing information to help focus the search, and performing exploratory data mining based on the intermediate data mining results. In addition, this component allows the user to browse database and data warehouse schemas or data structures, evaluate mined patterns, and visualize the patterns in different forms.

2. DATA SETS

Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases. Data mining is widely used domain for extracting trends or patterns from historical data. Dataset is a collection of data, usually presented in a tabular form. Each column represents a particular variable, and each row corresponds to a given member of the data. RDBMS has become a standard for storing and retrieving large amount of data. This data is permanently stored and retrieved through front end applications. The applications can use SQL to interact with relational databases. Preparing databases needs identification of relevant data and then normalizing the tables. Generally, data sets that are stored in a relational database or a data warehouse come from On-Line Transaction Processing (OLTP) systems in which database schemas are highly normalized. But data mining, statistical or machine learning algorithms generally require aggregated data in summarized form. Suitable data set building for a mining purposes is a time- consuming task. This task requires writing long SQL statements or customizing SQL Code if it is automatically generated by some tool. Thus in a relational database, a lots of effort is required to prepare a data set that can be used as input for a data mining or statistical algorithm. Most algorithms require data set as a input which is in horizontal form, with several records and one variable or dimension per column.

A relational database system produce normalized tables for analysis. To get more details for analysis, denormalized tables are used. With the help of SQL queries users can perform aggregation of tables and can produce results in vertical and horizontal layout. Preparing the useful and appropriate data set for data mining, needs more time. To convert the dataset into suitable form the data practitioner has to write the complex SQL queries. The two main operations that are used in such SQL queries are **join** and **aggregation**. The most well-known aggregation is the aggregating of a column over group of rows. In SQL, the aggregation of data is done using the aggregate functions such as minimum, maximum, average, count and sum and the result is obtained in the vertical layout. Each of these functions returns its result over a group of rows. But these aggregate functions are unable to prepare the suitable dataset for the data mining purposes. So a significant effort is needed for computing aggregation when they are required in horizontal layout. With such drawback in mind, a new class of aggregate methods that aggregates numeric expressions and transpose data to produce a data set with a horizontal layout. So the new set of functions called **horizontal aggregation** is used to get the dataset in horizontal layout. The three methods for evaluating horizontal aggregation are SPJ method, CASE method and PIVOT method.

3. AGGREGATIONS

Aggregations plays an vital role in SQL code. Aggregation is the grouping the values of multiple rows together to form a single value. There are mainly two types of aggregation techniques which include vertical aggregation and horizontal aggregation. There are various methods that are used for preparing horizontal layout for dataset. That can be used for the data mining analysis. Various approaches that are used in the data set preparation are classified as follows as shown in figure 2.

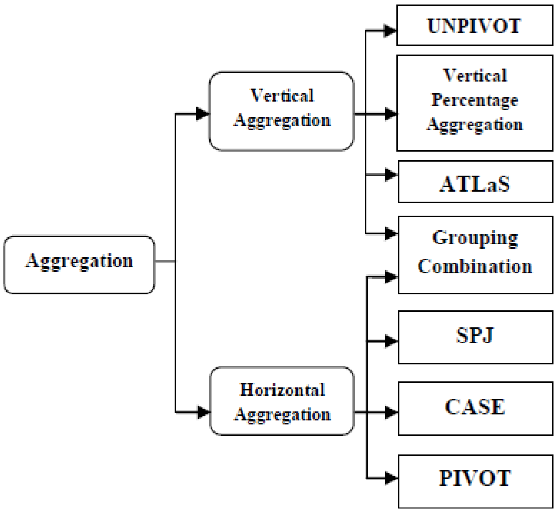

Fig2. Classification of Aggregation

3.1 Vertical aggregation

Existing SQL aggregations are also called as vertical aggregation. Vertical aggregations return single value. The most com- mom vertical aggregation functions includes sum(), count(), avg(), min(), etc. The vertical aggregations are common for numerous programming such as relational algebra. The output that we get from existing aggregations cannot be directly used for data mining algorithm. Data mining algorithm requires data in the form of data set. To get data in the form of data set from the output of existing aggregations require joining tables and columns , aggregating columns and many complex queries. It means that vertical aggregations have limitations to prepare data set.

3.2 Horizontal aggregation

Horizontal aggregations are also similar to standard SQL aggregations but this can produce results in horizontal tabular format. Horizontal aggregation returns set of values instead of single value. Horizontal aggregation returns the output in the form of horizontal layout or in summarized form. The output that we get from horizontal aggregation can be directly used for data mining. The limitation of vertical aggregation is overcome in horizontal aggregation. Horizontal aggregation is evaluated using three methods which includes CASE, SPJ and PIVOT method. Here data sets for all the operations are produced from some data mining tool and apply the aggregation operations on that dataset. To produce results in horizontal layout small syntax extensions to normal SQL syntax is needed. The syntax for horizontal aggregation is given below.

SELECT columns, Aggregation Function (Measure column BY Aggregating Parameters) FROM GROUPING columns

3.3 ADVANTAGES

In horizontal aggregations several advantages are there. The first advantage is horizontal aggregation represent a template to generate SQL code from a data mining tool. This SQL code reduces manual work in the data preparation phase in data mining related project. The Second is automatically generating the SQL code, which is more efficient than an end user written SQL code. The datasets for the data mining projects can be created in less time. The third advantage is the data sets can be created entirely inside the DBMS.

4. DATASET PREPARATION

Dataset is a collection of data, usually presented in a tabular form. Each column represents a particular variable, and each row corresponds to a given member of the data. Dataset preparation [1] is very important for any of the operations in the data mining analysis. Preparation of dataset addresses many issues and there are solutions for overcoming this problem. For performing operations on data stored inside the database system, users normally use SQL queries to retrieve those data. After retrieving perform various extractions and transformations on the dataset to make them suitable for application. Some approaches require denormalized table than normalized table. Because that contain more details than normalized tables and many analysis require analysis on large amount of data. RDBMS has become a standard for storing and retrieving large amount of data and generally, data sets that are stored in a relational database or a data warehouse come from On-Line Transaction Processing (OLTP) systems in which database schemas are highly normalized. The major issues in the creation and transformation of variables for analysis are selection of appropriate record from the available large dataset and preparation of efficient SQL queries to optimize them. Most the issues in the creation and transformation of dataset is related to summarization, aggregation, denormalization, cross-tabulation. Sometimes analysis need summarized details then detailed. So there is a need for summarized data. Cross-tabulation is also an important concept because it gives detailed information after analysis in a horizontal tabular format. Those are easy to understand than vertical format. Horizontal aggregation almost similar to normal aggregation but it uses some syntax extensions. Selecting SQL queries face some difficulties when using left outer join for effectiveness and use of appropriate SQL queries for each operation and handling of multiple primary keys.

Since dataset preparation is most expensive and time consuming task, thus dataset preparation is very important. The data is stored inside the database system can get the benefit of database management system. There are four steps for the dataset preparation. Dataset preparation start with data selection. In data selection, the analyst wants to perform analysis on the available data and select appropriate data for analysis. Second step is data integration. In data integration, data collected from different source are combined and stored inside a table. Third one is the data transformation. In data transformation the analyst wants to transform data into the format required for each operation. The last step is the data reduction. Here the data is compressed for the easiness of the analysis. As per this way there is a data preparation framework [2] for efficiently preparing dataset for analysis. Since the Quality of any analysis depends on the quality of data being processed.

5. COMPARATIVE STUDY

In data mining many approaches need tabular datasets for operations. Tabular datasets are easier to understand and use than any other approaches. Aggregations using SQL play a major role in producing tabular datasets. There are several data mining approaches that use SQL extensions for information extraction. For example, integration of association rule uses SQL implementations with the support of an a priory algorithm [3]. But the use of normal SQL queries has low performance than other architectures. So every approaches use some extensions on the normal SQL syntax to improve the performance. K-means clustering also use SQL implementations to get the benefit of DBMS [4]. There is another approach, Bayesian classifier also use SQL for operations. The major advantages of using DBMS are storage management, concurrency control, and security. Here various approaches for performing aggregation are taken for comparison.

5.1 UNPIVOT Operator UNPIVOT operator is also an aggregation operator for producing results in tabular format. This operator works in opposite of PIVOT operator that is they transform columns into rows. This creates additional rows from columns to produce a big table. Because of these vertical layout it cannot used for most of the mining algorithms which require horizontal table as input. UNPIVOT operator [5] is commonly used for the statistical computation of some data mining approaches. The normal syntax is given below. *SELECT columns FROM table UNPIVOT (Measure Column FOR Pivot Column IN (Pivot Column Values))*

5.2 ATLaS ATLaS is a database language developed to solve the limitations of SQL operator. ATLaS [6] can perform aggregations that are not possible with standard SQL. Standard SQL can support only basic aggregation operations. This language use aggregations and table functions in SQL. To perform operations in ATLAS entire SQL statement is divided into three functions INTIALIZE, ITERATE, TERMINATE. Declarations are given in the INITIALIZE section. The major operation is specified in the ITERATIVE section. The final statement to execute is specified in the TERMINATE sections. The major advantage of ATLaS is that it can support online aggregations. In online aggregation user evaluate aggregation query in an online fashion execution. But the execution of ATLaS operator consumes more space than executing with normal SQL. Also it cannot results in horizontal tabular format.

5.3 Vertical and Horizontal percentage aggregations

This aggregations help to calculate percentages for operations using vertical and horizontal aggregations [7]. Vertical percentage aggregation returns one row for percentage in vertical format. Horizontal percentage aggregation returns entire 100% of results on the same row. This percentage aggregation used only for computing percentages in vertical or horizontal format. These aggregations are similar to normal vertical and horizontal aggregation except that it can compute results only in percentage format. So there may be extra work in the percentage conversion when other computations are required on the dataset.

5.4 Grouping Combination GROUPING COMBINATION operator [8] is developed to handle the aggregation and grouping of high dimensional data. This operator can solve limitations of normal GROUPING operator. The operators like GROUPING SET, ROLLUP, and CUBE can also perform aggregation and can produce tabular results. But these are difficult to use when the available input dataset is very large. When the available input dataset is large GROUPING SET operator require long complex SQL queries. The ROLL UP operator can perform aggregation on smaller datasets and produce tabular results vertical format. But the vertical format is not efficient for many data mining approaches. The CUBE operator can perform aggregations on large datasets. But the CUBE operator eliminates some of the details when aggregation is performed. Because of these limitations GROUPING COMBINATION operator is developed. But the GROUPING COMBINATION operator can implemented only with the help of complex algorithms. So its performance is low in the case of execution.

5.5 SPJ Method In SPJ method [9] create one table with a vertical aggregation for each result column, and then join all those tables top produce FH. Here d projected tables are created from d select, project, join queries. Left outer join queries are used to join all the projected tables. SPJ method can produce tables in horizontal layout an optimized SPJ method can produce more efficient result. The performance of SPJ approach is very low when there is large number of rows. But this can perform aggregation with the help of basic SQL queries. This is easier to support by any database.

5.6 CASE Method CASE method can be performed by combining GROUP-BY and CASE statements [9]. It is more efficient and has wide applicability. CASE statement evaluates the Boolean expression and return value from the selected set of values. CASE statement put the result to NULL when there is no matching row is found. This also produce resultant table in a horizontal layout.

SELECT columns, Aggregate Function (CASE WHEN Boolean expression THEN result ELSE result expression END) FROM table GROUP BY columns

6. CONCLUSION

Since most of the data mining algorithms require datasets in horizontal layout. In this article we can see that aggregations is a challenging and interesting problem, since Aggregations plays an vital role in SQL code. Mainly, the existing SQL aggregations return results in one column per aggregated group. But in horizontal aggregation, it returns a set of numbers instead of a single number for each group. Here in this article various approaches for performing aggregation are presented. Among this most data mining algorithms require the methods which produce horizontal tabular datasets. This is because in horizontal layout each row contains more details instead of one number in a row. It produce resultant table with more column and few rows. Here there are three methods which produce horizontal table. This article propose a new class of extended aggregate functions, called horizontal aggregations which help preparing data sets for data mining and OLAP cube exploration by takes the advantage all the three methods(SPJ,CASE and PIVOT) in aggregation and uses them for data mining approaches.

REFERENCES

[1] C.Ordonez, "Data Set Preprocessing and Transformation in a Database System," *Intelligent Data Analysis*, vol. 15, no. 4, pp. 613-631, 2011.

[2] Kai-Uwe Sattler, Eike Schallehn, "A Data Preparation Framework based on a Multidatabase Language," *IEEE Trans. Knowledge and Data Eng*, 2001.

[3] S. Sarawagi, S. Thomas, and R. Agrawal, "Integrating Association Rule Mining with Relational Database Systems: Alternatives and Implications," *Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '98)*, pp. 343-354, 1998.

[4] C. Ordonez, "Integrating K-Means Clustering with a Relational DBMS Using SQL," IEEE Trans. Knowledge and Data Eng., vol. 18,no. 2, pp. 188-201, Feb. 2006.

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

[6] H. Wang, C. Zaniolo, and C.R. Luo, "ATLAS: A Small But Complete SQL Extension for Data Mining and Data Streams,"*Proc. 29th Int'l Conf. Very Large Data Bases (VLDB '03)*, pp. 1113-1116, 2003.

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

[8] Alexander Hinneburg, Dirk Wolfgang Lehner, "Combi-Operator-Database Support for Data Mining Applications," *Proc. 29th VLDB Conference*, 2003.

[9] C. Ordonez, "Horizontal Aggregations for Building Tabular Data Sets," *IEEE Trans. Knowledge and Data Eng*, VOL. 24, NO. 4, April 2012.

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

[11] Jennifer L. Beckmann, Alan Halverson, Rajasekar Krishnamurthy, Jeffrey F. Naughton, "Extending RDBMSs to Support Sparse Datasets Using An Interpreted Attribute Storage Format," *An enterprise directory solution with DB2. IBM Systems Journal*, 39(2), 2005.

[12] C. Ordonez, "Statistical Model Computation with UDFs," *IEEE Trans. Knowledge and Data Eng*., vol. 22, no. 12, pp. 1752-1765, Dec.2010.

[13] C. Ordonez, Zhibo Chen. Horizontal aggregations in SQL to prepare Data Sets for Data Mining Analysis. IEEE Transactions on Knowledge and Data Engineering (TKDE), 2012.