Bulletin of the Marathwada Mathematical Society
Vol.-5, N0.2, December 2004, Pages 38 - 52

DATA WAREHOUSING AND MINING AN INDISPENSABLE COMPUTATIONAL TOOL FOR REAL WORLD PROBLEMS

V. Sree Hari Rao
Department of Mathematics,
Jawaharlal Nehru Technological University,
Hyderabad – 500 072, India. E-mail:vshrao@yahoo.com
and
J.V.R. Murthy:
Department of Computer Science and Engineering,
J.N.T.U.College of Engineering,
Kakinada, India. E-mail:murthyjonnalagedda@yahoo.com

ABSTRACT

The main objective of this paper is to convey the readers that Data Warehousing and Mining is an indispensable tool for constructing relevant Decision Support System. This is a very upcoming area that has high bearing in solving problems arising out of globalization and liberalization process of Govt. of India and WTO. In this paper the current day scenario, evolution and growth of the computers is discussed from this point of view. The concepts of Data Warehousing are discussed with a real time Health Insurance problem. Subsequently different Data Mining techniques and their applications are discussed. A practical application of clustering is given along with C language code.

Key Words: Data Warehousing, Data Mining, ERP, OLTP, Cluster and Decision Tree. Decision Support System (DSS).

1. INTRODUCTION

Today in the fast growing competitive corporate world, quick, reliable and sustainable business solutions is the essentiality of the hour. A close look at the present scenario tells that various applications are spread across heterogeneous platforms. For example a corporate company might be implementing Entrepreneur Resource Planning package for efficient and faster access of data. It might be using People Soft for human resource management system, pay roll and benefit applications, SAP for material management, sales, distribution supply chain, and ORACLE financials for general ledger and other financial applications.

All branches of the office spread across the globe are networked through satellite communication links or fiber optic cables. Data can be accessed and transferred from one place to another in no time. If we look into the past, in yester years computers were used for running some batch processing jobs such as payroll and inventory control etc. The Data was initially stored on punch cards and subsequently transferred on to the magnetic tapes, a serial device. The major bottleneck with this is, searching takes a great deal of time and online processing is a difficult task. It is a big hurdle for business applications. Subsequently random accesses such as floppies and hard disk were invented and improved and they have facilitated on line processing. Hence many real time applications such as banking, stock market and industrial process control had a faster growth. With conventional file processing there were some problems like redundancy, security, and data integrity. One important issue is that queries cannot be answered on the spot just by merely mentioning what is needed. The concept of database was introduced to address the above said problems. The relational database model has sound mathematical foundation.

2. EVOLUTION OF DATABASE SYSTEMS

Since the 1960s, database and information technology has been evolving and systematically from primitive file processing systems to sophisticated and powerful database systems ([1]). The research and development in database systems since the 1970s has progressed from early hierarchical and network database systems to the development of relational database systems, where data are stored in relational table structures, data modeling tools, and indexing and data organization techniques. In addition, users gained convenient and flexible data access through query languages, user interfaces; optimized query processing and transaction management. Efficient methods for on line transaction processing (OLTP), where a query is viewed as a read only transaction, have contributed substantially to the evaluation and wide acceptance of relational technology as a major tool for efficient storage, retrieval, and management of large amounts of data.

Database technology since the mid -1980s has been characterized by the popular adoption of relational technology and an upsurge of research and development of activities on new and powerful database systems. These employ advance data models such as extended-relational, object-oriented, object-relational, and deductive models. Application-oriented database systems, including spatial, temporal, multimedia, active, and scientific databases knowledge bases and office information bases, have flourished. Issues related to distribution, diversification, and sharing of data have based the global information systems such as World Wide Web (WWW) which have also emerged and played a vital role in the information industry. The steady and amazing progress of computer hardware technology in the past three decades has led to large supplies of powerful and affordable computers, data collection equipment, and storage media. This technology provides a great boost to database and information industry, and makes a large number of databases and information repositories available for transaction management, information retrieval and data analysis.

The current issue is to predict the production of the corporate company for the next quarter. In order to achieve this, it is required to collect relevant information from HRMS, material management, sales distribution and financial modules. It essentially requires historical data. Then only the prediction will be accurate. Some times the data requires some consolidation, cleaning, normalization and preprocessing. Such data obtained after all the transformations is loaded into the "Data warehouse". This data warehouse needs to be mined for interesting patterns, from which DSS evolves.

3. DATA WAREHOUSE

According to W.H.Inmon ([2]), a leading architect in the construction of data warehouse systems," A data warehouse is a subject-oriented, integrated, time-variant, nonvolatile collection of data in support of management's decision making process". This short, comprehensive definition presents the major features of a data warehouse. The four key words, subject-oriented, integrated, time-variant, and nonvolatile distinguish data warehouses from other data repository systems, such as relational data base systems, transaction processing systems, and file systems. Let us take a close look at each of these key features. ([3] [4] [5] [6])

Subject oriented: A data warehouse is organized around major subjects such as customer, supplier, product and sales. Rather than day-to-day operations and transaction processing of an organization, a data warehouse focuses on the modeling and analysis of data for decision makers. Hence, data warehouses typically provide a simple and concise view around particular subject issues by excluding those not useful in the decision support process.

***Integrated**:* A data warehouse is usually constructed by integrating multiple heterogeneous sources, such as relational databases, flat files and on-line transaction records. Data cleaning and data integration techniques are applied to ensure consistency to naming conventions, encoding structures, attribute measures, and so on.

Time-variant: Data are stored to provide information from historical perspective (e.g., the past five-ten years). Every key structure in the data warehouse contains, either implicitly or explicitly, an element of time.

Nonvolatile: A data warehouse is always a physically separate store of data transformed from the application data found in the operational environment. Due to this separation, a data warehouse does not require transaction processing, recovery and concurrency control mechanisms. It usually requires two operations in data accessing such as *Initial loading of data and access of data*. However it is agreed by researchers and vendors alike that data warehouses are built in the interest of business decision support and contain historical data summarized and consolidated from detailed individual records from a number of operational databases.

4. A REAL WORLD EXAMPLE

We now consider a real world example for understanding the warehouse loading methodology. Health insurance providers offer insurance to various clients. A corporate employer, offering insurance benefits to his employees such as group insurance, requires studying the response of clients to various policies. Hence the provider asks various clients to send the relevant information needed for a decision support system. The clients will send the data periodically. Then the data will be further consolidated depending on the situation. The following attribute values are expected from the client database. Record ID, personal identification number (PIN), name, date of birth, sex, marital status, address, coverage end date, date

of hiring, benefit plan, coverage begin date, new employee indicator, terminated employee indicator, employee coverage indicator, spouse coverage indicator, child coverage indicator. The provider collects data from various clients and merges those flat files (an ASCII file with above attributes) and consolidates them to load the data warehouse. Now let us closely look at the extraction process. In the above mentioned fields, some fields can be extracted from appropriate tables, where as the fields like coverage indicators need some process to be executed. The above extraction can be done using conventional methods, which consume moretime. Instead the same could be achieved through creating metadata structures and temporary arrays more efficiently.

Some of the important tables involved in this process are given below.

- Personal-data: A table that describes the personal information of all the employees.
- Employment: The employment particulars of employees.
- Job: The service register, which stores all the service related transactions.
- Health-benefit: This table stores information on every employee and health plan enrollments from the date of joining to as of date.
- Health-dependent: This table is a child table to health benefit table and stores the dependent enrollment information.

In the above mentioned tables job, health benefits and health dependent tables consists of large number of records. They are dynamic tables in the sense that the records in the table get updated continually with more insertions. So joining such tables is an expensive task.

In the conventional method Health-benefits and Health-dependent tables will be joined. The information pertaining to the employees and dependents, participating in benefit plans offered by a specific provider and terminated employee records are extracted. For each terminated employee it is required to check the previous active row to decide whether the person was with same provider or not. In the terminator record the provider information will not be available. Hence it is required to examine the previous row. The following issues reduce the performance at this stage.

- The joining of two dynamic tables:
- Obtaining all terminated rows and going back to previous active row and checking for the provider.

So far as coverage indicator extraction is considered, there is one more bottleneck. To get the indicators it is required to join various tables repeatedly. Each time an operation is performed, the database needs to be accessed resulting in the increased network traffic.

The three above said bottlenecks can be resolved in the new model, which follows the data warehouse methodology. In this case joining of two tables and obtaining all the terminated rows and filtering the irrelevant rows may be avoided by a metadata structure that is a temporary table (temp table). The temp table consists of previous run data and health benefits consist of current data. Hence, by comparison terminates and new recruiters can be easily obtained. By extracting the data of an employee and dependents into an array and processing, it will reduce the network traffic and repeated access of the same tables. We refer the readers to ([7]) for further details on the structured query language statements and the algorithms.

5. DATA MINING.

Until now, we discussed the issues related to Data Warehousing. Now, let us consider what “Data Mining” means and how data warehousing and mining are helpful for a decision support system ([8]). Decision Support Systems [DSS] are comprehensive computer systems and related tools that assist managers in making decisions and solving problems. The goal is to improve the decision-making process by providing specific information needed by management. These systems defer from traditional database management systems, in that, more adhoc queries and customized information may be provided. Recently, the terms Executive Information System (EIS) and Executive Support System (ESS) have developed as well. All these systems aim at developing the business structure and computer techniques to better provide information needed by management to make effective business decisions. Data Mining can be thought of as a suit of tools that assist in the overall DSS process i.e., DSS may use Data Mining tools.

In many ways the term DSS is much broad than the term data mining. While a DSS usually contains data mining tools, this need not to be so. Likewise, a data mining tool need not be contained in a DSS system. A decision support system could be enterprise-wide, thus allowing upper-level managers the data needed to make intelligent business decisions that impact the entire company. A DSS typically operates using data warehouse data. Alternatively, a DSS could be build around a single user and a PC. The bottom line is that a DSS gives managers the tools needed to make intelligent decisions.

Simply stated, Data Mining refers to extracting or “Mining” knowledge from large amounts of data. ([4] [9] [10] [11]) The major reason that Data Mining has attracted a great deal of attention in the information industry in recent years is due to the wide availability of huge amounts of data and the imminent need for turning such data into useful information and knowledge. The information and knowledge gained may be used for applications ranging from business management, production control and market analysis, to Engineering Design and Science Exploration.

To the newcomer, the plethora of different approaches initially can be quite confusing, and the situation is not helped by the inconsistent terminology used among data mining practitioners themselves. Although no single set of terms enjoys universal approval, we offer a frame of reference in the figure (given on next page). One may use the figure as a starting point as well as a graphical table of contents for the rest of this chapter.

The applications listed in the Figure on the next page represent typical business applications where data mining is used today.

5.1 Data Mining Operations

Predictive modeling, database segmentation, link analysis, and deviation detection are the four major operations for implementing any of the business applications. We deliberately do not show a fixed, one-to-one link between the business applications and data mining operation layers, to avoid the suggestion that only certain operations are appropriate for certain applications and vice versa. On the contrary, really breakthrough results may sometimes come from the use of non-intuitive approaches to problems. Nevertheless, certain well-established links between the applications and the corresponding operations do exist. For example, modern target marketing strategies are almost always implemented by means of the database segmentation operation. However, fraud detection could be implemented by any of the four operations, depending on the nature of the problem and input data. Furthermore, the operations are not mutually exclusive. For example, a common approach to customer retention is to segment the database first and then apply predictive modeling to the resultant, more homogeneous segments. Typically the data analyst, perhaps in conjunction with the business analyst, selects the data mining operations to use

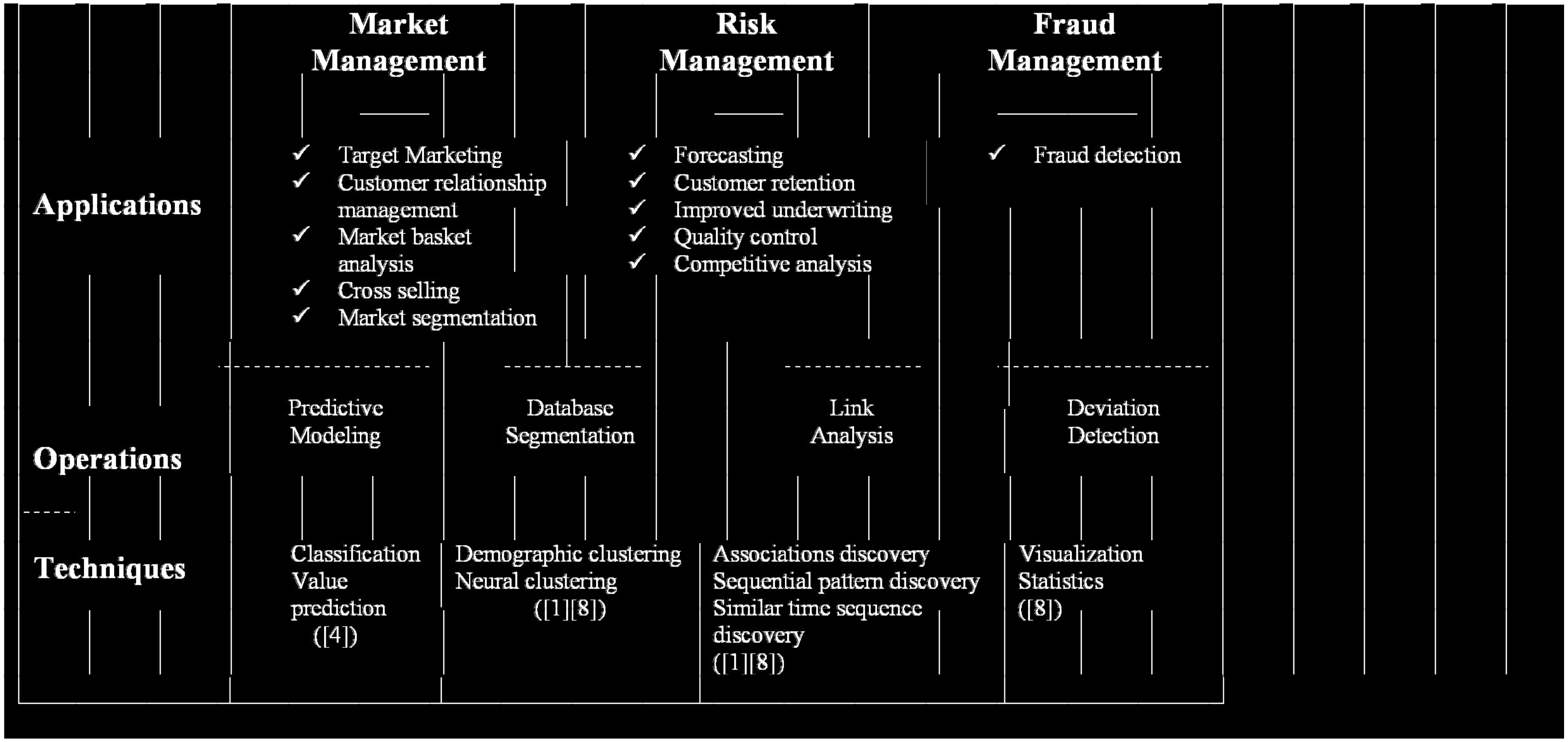

Figure: Data Mining Applications and Their Supporting Operations and Techniques

The Data Mining may be viewed as an essential step in the process of knowledge discovery in databases. Knowledge discovery process consists of the following steps:

- Data cleaning (to remove noise and inconsistent data): One approach to cleaning data is to simply delete cases that contain missing values. However, the results are biased because the deleted data may have otherwise been an important part of various relationships. Major cleaning such as making variable names consistent, inputting missing values, identifying errors, correcting errors, and detecting outliers can be performed relatively easily using Data Mining.
- Data integration (where multiple data sources may be combined) : A popular trend in the information industry is to perform data cleaning and data integration as a pre-processing step where the resulting data are stored in a data warehouse..

- Data selection (where data relevant to the analysis task are retrieved from the data base).

- Data transformation (where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations, for instance.): Some times data transformation and consolidation are performed before the data selection process, particularly in the case of data warehousing.

- Data mining: An essential process where intelligent methods are applied in order to extract data patterns.

- Pattern evaluation: To identify the truly interesting patterns representing knowledge based on some interesting business measures.

- Knowledge presentation: Where visualization and knowledge representation techniques are used to present the knowledge to the user.

5.2 Data Mining functionalities

We now have a look at the data mining functionalities briefly.

Data mining functionalities are used to specify the kind of patterns to be found in Data Mining tasks. In general, Data Mining Tasks can be classified into two categories: Descriptive and Predictive. Descriptive mining tasks characterize the general properties of the data in the database. Predictive mining tasks perform inference on the current data in order to make valid predictions.

Concept/class description: Data can be associated with classes or concepts. For example, in a departmental store, classes of items for sale include rice, wheat and bread and concepts of customers include luxury spenders and budget spenders. It is useful to describe individual classes and concepts in summarized, concise, and yet precise terms. Such descriptions of a class or a concept are called Class/Concept descriptions. These descriptions can be derived via (1) Data characterization, by summarizing the data of the class under study (often called the target class), in general terms, or (2) Data discrimination, by comparison of the target class with one or a set of comparative classes (often called contrasting classes) or (3) both data characterization and discrimination.

Association Analysis*:* This is the discovery of Association rules showing attribute value conditions that occur frequently together in a set of data.

Classification and prediction*:* Classification is the process of finding a set of models (or functions) that describe and distinguish data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown. The derived model is based on the analysis of a set of training data.

Cluster analysis*:* Unlike classification and prediction, which analyze class-labeled data objects, clustering analyses data objects without consulting a known class label. In general, the class labels are not present in the training data simply because they are not known to begin with. Clustering can be used to generate such labels.

Outlier analysis*:* a database may contain data objects that do not comply with the general behavior or model of the data. These data objects are outliers. Most data mining methods discard outliers as noise or exceptions.

However, in some applications such as fraud detection, the rare events can be more interesting than the regularly occurring ones. The analysis of outlier data is referred to as Outlier mining.

Evolution analysis: Data evolution analysis describes and models regularities or trends for objects whose behavior changes over time. Although this may include characterization, discrimination, association, classification, or clustering of time-related data, distinct features of such an analysis include time-series data analysis, sequence or periodicity pattern matching, and similarity-based data analysis.

5.3 An Example for Data Mining

We now consider an example for knowing the importance of Data Mining. A premier educational institution wants to send its topper to national level competitive examinations. The student class marks, assignment marks and external exam marks are available in the data base tables. They are all consolidated and the table structure is like the following:

Student_info = (rollno, name, maths, physics, chemistry).

The table is in the sorted order of roll numbers. Consider a set of fifteen records for our study. The actual database consists of hundreds of thousands of records.

2004CSE501J.V.R.MURTHY	**099000960008500**
2004CSE502S.RAVISANKAR	089000855007500
2004CSE503G.VAMSHIKISHORE	086000895008600
2004CSE504T.SRICHARAN	085000915008400
2004CSE505P.NARAYANA	078000860008200
2004CSE506T.SRIVARDHAN	**098000895008600**
2004CSE507A.NAGACHAITANYA	083000780007800
2004CSE508M.KIRANKUMAR	084000850008200
2004CSE509P.RAGHAVA	067000760007800
2004CSE510M.RAJESH	079000755007200
2004CSE511A.VISHWANATH	069000680007200
2004CSE512C.MURALIKRISHNA	074000690007300
2004CSE513K.RAJASHEKAR	068000630007600

2004CSE514J.KAMALHASAN	082000865008300
2004CSE515K.JEEVANKUMAR	081000805008200

Out of these decide: who should be picked up? Obviously, roll numbers 501 and 506. Let us consider the Data Mining methodology for their selection. The normal way of selection is to sort the table on marks and obtain the toppers in the range of 95 to 100. Sorting has a time-complexity of **(n x n)** where n is the number of records. If clustering technique is used, it has a time complexity of (k x n x t)where k is the number of clusters, n is the number of records and t is the number of transactions (the number of transfers of records among various clusters is the number of transactions). In practice, k and t are much smaller compared to n. Hence, clustering consumes much less time than sorting. Moreover, clustering is much more suitable if we have weights for different attributes and when various distance calculation measures are included.

The following C – program groups the data into three clusters. The cluster one consists of two records with roll numbers 501 and 506. So, our guess is correct!

The k-mean clustering algorithm

Input: The file containing student records, n (no. of clusters)
Output: The clusters

```
begin main_procedure
    open(student_file);
    set i=0
    while(i<n)
    {

    generate a random number r;
        extract the r th  record from the file;
        check the existence of this record as cluster centre
        if (not exists previously)
            make it the centre for cluster I;
            i++;
        else
```

```
                    skip
        } end-while
        rewind the student_file;

        while(not eof)
        {
                read record;
                create a node for the record;
                c = call procedure get_cluster_num
                call procedure add_node(c);
        } end-while
close(student_file);
        call procedure realignment();
end-procedure


procedure get_cluster_num()
begin-procedure
        read the values from the node;
        for(i=0;i<n;i++)
get the distance between node and cluster(i);
                store it in an array;
        end-for
        get the minimum distance and the cluster(i);
        return()
end-procedure
procedure realignment()
begin procedure
        set the transfer_flags array to zeros
        set change_flag='y';
        set j=0  /* The list to be considered */
        while (change=='y')
        {
```

```
        while(not end of list(j))
        {
        e1=call_procedure get_cluster_num();
        if (c1!= j)      /* The element should be transferred from j to c1*/
            transfer the element from j to c1
            set Transfer_flag[i]=1;
            set Transfer_flag[c1]=1;
            set move_flag=0;
        else
            move to the next node of jth cluster
        end-if
} end-while
        if (move_flag=1)
            set transfer_flag[j]=0;
        end-if
        set move_flag=0;
        set j-> j+1;
        if(j = = n )
            set j=0;
        end-if
        get change_flag
    } end-while
end-procedure
```

Thus Mathematics (especially Statistics and Fuzzy logic) and Neural Networks have an excellent role to play in Data Mining. In summary, the real world problems that have a bearing on very large data needs a special attention and our above presentation aptly suggests that Data Warehousing and Mining would thus be an indispensable computational tool for constructing relevant Decision Support Systems.

REFERENCES

[1] **Han, J.** and **Kamber, M.,** *"Data Mining: Concepts and Techniques",* Morgan & Kaufmann, 2000.

[2] **Imman, W.H.,** *"Building the Data Warehouse",* John Wiley & Sons, New York,

[3] **Kimball, R.,** *"The Data Warehouse Tool Kit",* John Wiley & Sons, New York.

[4] **Pujari, Arun K.,** *"Data Mining Techniques"*,Universities Press, Hyderabad, India, .

[5] **Chaudhury, S.** and **Dayal, U.** *An overview of data warehousing and OLAP technology,* SIGMOD Record **26(1)**(1997), 65-74.

[6] **Anhory, S.** and **Murray, D.,** *"Data warehousing in the Real world: A Practical Guide for Building Decision Support Systems",* Addison Wesley Longman, 1997.

[7] **Murthy Jonnalagedda, Sarvarayudu, G.P.R.** and **Ananda Mohan, K.,** *Performance tuning the health insurance benefits extract, Proceedings of 6th International congress on Insurance – Mathematics and Economics;* July 2002,Techincal University, Lisbon Portugal.

[8] **Margaret H.Dunham**, *Data Mining Introductory and advanced topics,* Pearson Education, Singapore, 2004.

[9] **Barbara, D.** (Ed.), *Special issue on "Mining of Large Databases",* IEEE Data Engineering Bulletin **21(1)**(1998).

[10] **Heckerman, D.,** *Bayesian networks for data mining, Proceedings of the 6th International Conference on Database Theory,* Springer LNCS 1186, 1997.

[11] **Piatetsky – Shapiro, G.,** and **Frawley, W.,** (eds): *Knowledge Discovery in Databases,* MIT Press, Cambridge, Ma, 1991.