Database Queries, Data Mining, and OLAP

Lutz Hamel
University of Rhode Island, USA

INTRODUCTION

Modern, commercially available relational database systems now routinely include a cadre of data retrieval and analysis tools. Here we shed some light on the interrelationships between the most common tools and components included in today's database systems: query language engines, data mining components, and on-line analytical processing (OLAP) tools. We do so by pair-wise juxtaposition which will underscore their differences and highlight their complementary value.

BACKGROUND

Today's commercially available relational database systems now routinely include tools such as SQL database query engines, data mining components, and OLAP (Craig, Vivona, & Bercovitch, 1999; Hamm, 2007; Melomed, Gorbach, Berger, & Bateman, 2006; Scalzo, 2003; Seidman, 2001). These tools allow developers to construct high powered business intelligence (BI) applications which are not only able to retrieve records efficiently but also support sophisticated analyses such as customer classification and market segmentation. However, with powerful tools so tightly integrated with the database technology understanding the differences between these tools and their comparative advantages and disadvantages becomes critical for effective application development. From the practitioner's point of view questions like the following often arise:

- Is running database queries against large tables considered data mining?
- Can data mining and OLAP be considered synonymous?
- Is OLAP simply a way to speed up certain SQL queries?

The issue is being complicated even further by the fact that data analysis tools are often implemented in terms of data retrieval functionality. Consider the data mining models in the Microsoft SQL server which are implemented through extensions to the SQL database query language (e.g. predict join) (Seidman, 2001) or the proposed SQL extensions to enable decision tree classifiers (Sattler & Dunemann, 2001). OLAP cube definition is routinely accomplished via the data definition language (DDL) facilities of SQL by specifying either a star or snowflake schema (Kimball, 1996).

MAIN THRUST OF THE CHAPTER

The following sections contain the pair wise comparisons between the tools and components considered in this chapter.

Database Queries vs. Data Mining

Virtually all modern, commercial database systems are based on the relational model formalized by Codd in the 60s and 70s (Codd, 1970) and the SQL language (Date, 2000) which allows the user to efficiently and effectively manipulate a database. In this model a database table is a representation of a mathematical relation, that is, a set of items that share certain characteristics or attributes. Here, each table column represents an attribute of the relation and each record in the table represents a member of this relation. In relational databases the tables are usually named after the kind of relation they represent. Figure 1 is an example of a table that represents the set or relation of all the customers of a particular store. In this case the store tracks the total amount of money spent by its customers.

Relational databases do not only allow for the creation of tables but also for the manipulation of the tables and the data within them. The most fundamental operation on a database is the query. This operation enables the user to retrieve data from database tables by asserting that the retrieved data needs to fulfill certain criteria. As an example, consider the fact that the store owner might be interested in finding out which customers spent more than $100 at the store. The fol-

Figure 1. A relational database table representing customers of a store

Id	*Name*	*ZIP*	*Sex*	*Age*	*Income*	*Children*	*Car*	*Total Spent*
5	Peter	05566	M	35	$40,000	2	Mini Van	$250.00
…	…	…	…	…	…	…	…	…
22	Maureen	04477	F	26	$55,000	0	Coupe	$50.00

lowing query returns all the customers from the above customer table that spent more than $100:

```
SELECT * FROM CUSTOMER_TABLE WHERE
TOTAL_SPENT > $100;
```

This query returns a list of all instances in the table where the value of the attribute *Total Spent* is larger than $100. As this example highlights, queries act as filters that allow the user to select instances from a table based on certain attribute values. It does not matter how large or small the database table is, a query will simply return all the instances from a table that satisfy the attribute value constraints given in the query. This straightforward approach to retrieving data from a database has also a drawback. Assume for a moment that our example store is a large store with tens of thousands of customers (perhaps an online store). Firing the above query against the customer table in the database will most likely produce a result set containing a very large number of customers and not much can be learned from this query except for the fact that a large number of customers spent more than $100 at the store. Our innate analytical capabilities are quickly overwhelmed by large volumes of data.

This is where differences between querying a database and mining a database surface. In contrast to a query which simply returns the data that fulfills certain constraints, data mining constructs models of the data in question. The models can be viewed as high level summaries of the underlying data and are in most cases more useful than the raw data, since in a business sense they usually represent understandable and actionable items (Berry & Linoff, 2004). Depending on the questions of interest, data mining models can take on very different forms. They include decision trees and decision rules for classification tasks, association rules for market basket analysis, as well as clustering for market segmentation among many other possible models. Good overviews of current data mining techniques and models can be found in (Berry & Linoff, 2004; Han & Kamber, 2001; Hand, Mannila, & Smyth, 2001; Hastie, Tibshirani, & Friedman, 2001).

To continue our store example, in contrast to a query, a data mining algorithm that constructs decision rules might return the following set of rules for customers that spent more than $100 from the store database:

```
IF AGE > 35 AND CAR = MINIVAN THEN
TOTAL SPENT > $100
```

or

```
IF SEX = M AND ZIP = 05566 THEN TOTAL
SPENT > $100
```

These rules are understandable because they summarize hundreds, possibly thousands, of records in the customer database and it would be difficult to glean this information off the query result. The rules are also actionable. Consider that the first rule tells the store owner that adults over the age of 35 that own a mini van are likely to spend more than $100. Having access to this information allows the store owner to adjust the inventory to cater to this segment of the population, assuming that this represents a desirable cross-section of the customer base. Similar with the second rule, male customers that reside in a certain ZIP code are likely to spend more than $100. Looking at census information for this particular ZIP code the store owner could again adjust the store inventory to also cater to this population segment presumably increasing the attractiveness of the store and thereby increasing sales.

As we have shown, the fundamental difference between database queries and data mining is the fact that in contrast to queries data mining does not return raw data that satisfies certain constraints, but returns models of the data in question. These models are attrac-