Mining Association Rules on a NCR Teradata System

Soon M. Chung
Wright State University, USA

Murali Mangamuri
Wright State University, USA

INTRODUCTION

Data mining from relations is becoming increasingly important with the advent of parallel database systems. In this paper, we propose a new algorithm for mining association rules from relations. The new algorithm is an enhanced version of the SETM algorithm (Houtsma & Swami 1995), and it reduces the number of candidate itemsets considerably. We implemented and evaluated the new algorithm on a parallel NCR Teradata database system. The new algorithm is much faster than the SETM algorithm, and its performance is quite scalable.

BACKGROUND

Data mining, also known as knowledge discovery from databases, is the process of finding useful patterns from databases. One of the useful patterns is the association rule, which is formally described in Agrawal, Imielinski, and Swami (1993) as follows: Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of items. Let D represent a set of transactions, where each transaction T contains a set of items, such that $T \subseteq I$. Each transaction is associated with a unique identifier, called transaction identifier (TID). A set of items X is said to be in transaction T if $X \subset T$. An association rule is an implication of the form X => Y, where $X \subset I$, $Y \subset I$ and $X \cap Y = \varnothing$. The rule X => Y holds in the database D with confidence c if c% of the transactions in D that contain X also contain Y. The rule X => Y has a support s if s% of the transactions in D contain X U Y. For example, beer and disposable diapers are items such that beer => diapers is an association rule mined from the database if the co-occurrence rate of beer and disposable diapers (in the same transaction) is not less than the minimum support, and the occurrence rate of diapers in the transactions containing beer is not less than the minimum confidence.

The problem of mining association rules is to find all the association rules that have support and confidence greater than or equal to the user-specified minimum support and minimum confidence, respectively. This problem can be decomposed into the following two steps:

1. Find all sets of items (called itemsets) that have support above the user-specified minimum support. These itemsets are called frequent itemsets or large itemsets.
2. For each frequent itemset, all the association rules that have minimum confidence are generated as follows: For every frequent itemset f, find all non-empty subsets of f. For every such subset a, generate a rule of the form a => (f - a) if the ratio of support(f) to support(a) is at least the minimum confidence.

Finding all the frequent itemsets is a very resource-consuming task, but generating all the valid association rules from the frequent itemsets is quite straightforward.

There are many association rule-mining algorithms proposed (Agarwal, Aggarwal & Prasad, 2000; Agrawal, Imielinski & Swami, 1993; Agrawal & Srikant, 1994; Bayardo, 1998; Burdick, Calimlim & Gehrke, 2001; Gouda & Zaki, 2001; Holt & Chung, 2001, 2002; Houtsma & Swami, 1995; Park, Chen & Yu, 1997; Savasere, Omiecinski & Navathe, 1995; Zaki, 2000). However, most of these algorithms are designed for data stored in file systems. Considering that relational databases are used widely to manage the corporation data, integrating the data mining with the relational database system is important. A methodology for tightly coupling a mining algorithm with relational database using user-defined functions is proposed in Agrawal and Shim (1996), and a detailed study of various architectural alternatives for coupling mining with database systems is presented in Sarawagi, Thomas, and Agrawal (1998).

The SETM algorithm proposed in Houtsma and Swami (1995) was expressed in the form of SQL queries. Thus, it can be applied easily to relations in the relational databases and can take advantage of the functionalities provided by the SQL engine, such as the query optimization, efficient execution of relational algebra operations, and indexing. SETM also can be implemented easily on a

parallel database system that can execute the SQL queries in parallel on different processing nodes. By processing the relations directly, we can easily relate the mined association rules to other information in the same database, such as the customer information.

In this paper, we propose a new algorithm named Enhanced SETM (ESETM), which is an enhanced version of the SETM algorithm. We implemented both ESETM and SETM on a parallel NCR Teradata database system and evaluated and compared their performance for various cases. It has been shown that ESETM is considerably faster than SETM.

MAIN THRUST

NCR Teradata Database System

The algorithms are implemented on an NCR Teradata database system. It has two nodes, where each node consists of 4 Intel 700MHz Xeon processors, 2GB shared memory, and 36GB disk space. The nodes are interconnected by a dual BYNET interconnection network supporting 960Mbps of data bandwidth for each node. Moreover, nodes are connected to an external disk storage subsystem configured as a level-5 RAID (Redundant Array of Inexpensive Disks) with 288GB disk space.

The relational DBMS used here is Teradata RDBMS (version 2.4.1), which is designed specifically to function in the parallel environment. The hardware that supports Teradata RDBMS software is based on off-the-shelf Symmetric Multiprocessing (SMP) technology. The hardware is combined with a communication network (BYNET) that connects the SMP systems to form Massively Parallel Processing (MPP) systems, as shown in Figure 1 (NCR Teradata Division, 2002).

The versatility of the Teradata RDBMS is based on virtual processors (vprocs) that eliminate the dependency on specialized physical processors. Vprocs are a set of software processes that run on a node within the multitasking environment of the operating system. Each vproc is a separate, independent copy of the processor software, isolated from other vprocs but sharing some of the physical resources of the node, such as memory and CPUs (NCR Teradata Division, 2002).

Vprocs and the tasks running under them communicate using the unique-address messaging, as if they were physically isolated from one another. The Parsing Engine (PE) and the Access Module Processor (AMP) are two types of vprocs. Each PE executes the database software that manages sessions, decomposes SQL statements into steps, possibly parallel, and returns the answer rows to the requesting client. The AMP is the heart of the Teradata RDBMS. The AMP is a vproc that performs many database and file-management tasks. The AMPs control the management of the Teradata RDBMS and the disk subsystem. Each AMP manages a portion of the physical disk space and stores its portion of each database table within that disk space, as shown in Figure 2 (NCR Teradata Division, 2002).

SETM Algorithm

The SETM algorithm proposed in (Houtsma & Swami, 1995) for finding frequent itemsets and the corresponding SQL queries used are as follows:

```
// SALES = <trans_id, item>
k := 1;
sort SALES on item;
F1 := set of frequent 1-itemsets and their counts;
R1 := filter SALES to retain supported items;
repeat
    k := k + 1;
    sort Rk-1 on trans_id, item1, . . . , itemk-1;
    R'k := merge-scan Rk-1, R1;
    sort R'k on item1, . . . , itemk ;
    Fk := generate frequent k-itemsets from the sorted R'k;
    Rk := filter R'k to retain supported k-itemsets;
until Rk = {}
```

In this algorithm, initially, all frequent 1-itemsets and their respective counts (F_1=<item, count>) are generated by a simple sequential scan over the SALES table. After creating F_1, R_1 is created by filtering SALES using F_1. A merge-scan is performed for creating R'_k table using R_{k-1}

Figure 1. Teradata system architecture

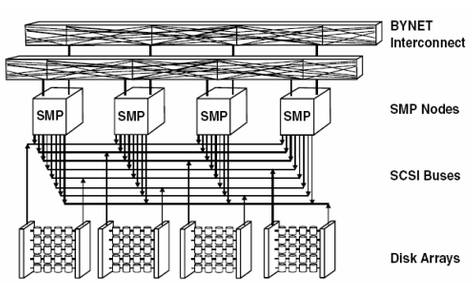

Figure 2. Query processing in the Teradata system