Mining Quantitative and Fuzzy Association Rules

Hong Shen
Japan Advanced Institute of Science and Technology, Japan

Susumu Horiguchi
Tohoku University, Japan

INTRODUCTION

The problem of mining association rules from databases was introduced by Agrawal, Imielinski, & Swami (1993). In this problem, we give a set of items and a large collection of transactions, which are subsets (baskets) of these items. The task is to find relationships between the occurrences of various items within those baskets. Mining association rules has been a central task of data mining, which is a recent research focus in database systems and machine learning and shows interesting applications in various fields, including information management, query processing, and process control.

When items contain quantitative and categorical values, association rules are called *quantitative association rules*. For example, a quantitative association rule derived from a regional household living standard investigation database has the following form:

$$age \in [50,55] \wedge married \rightarrow house.$$

Here, the first item, age, is numeric, and the second item is categorical. Categorical attributes can be converted to numerical attributes in a straightforward way by enumerating all categorical values and mapping them to numerical values.

An association rule becomes a fuzzy association rule if its items contain probabilistic values that are defined by fuzzy sets.

BACKGROUND

Many results on mining quantitative association rules can be found in Li, Shen, and Topor (1999) and Miller and Yang (1997). A common method for quantitative association rule mining is to map numerical attributes to binary attributes, then use algorithms of binary association rule mining. A popular technique for mapping numerical attributes is to attribute discretization that converts a continuous attribute value range to a set of discrete intervals and then map all the values in each interval to an item of binary values (Dougherty, Kohavi, & Sahami, 1995).

Two classical discretization methods are equal-width discretization, which divides the attribute value range into N intervals of equal width without considering the population (number of instances) within each interval, and equal-depth (or equal-cardinality) discretization, which divides the attribute value range into N intervals of equal populations without considering the similarity of instances within each interval. Examples of using these methods are given in Fukuda, Morimoto, Morishita, and Tokuyama (1996); Miller and Yang (1997); and Srikant and Agrawal (1996).

To overcome the problems of sharp boundaries (Gyenesei, 2001) and expressiveness (Kuok, Fu, & Wong, 1998) in traditional discretization methods, methods for mining fuzzy association rules were suggested. Many traditional algorithms, such as Apriori, MAFIA, CLOSET, and CHARM, were employed to discover fuzzy association rules. However, the number of fuzzy attributes is usually at least double the number of attributes; therefore, these algorithms require huge computational times. To reduce the computation cost of association mining, various parallel algorithms based on count, data, and candidate distribution, along with other suitable strategies, were suggested (Han, Karypis, & Kumar, 1997; Shen, Liang, & Ng, 1999; Shen, 1999a; Zaki, Parthasarathy, & Ogihara, 2001). Recently, a parallel algorithm for mining fuzzy association rules, which divides the set of fuzzy attributes into independent partitions based on the natural independence among fuzzy sets defined by the same attribute, was proposed (Phan & Horiguchi, 2004b).

MAIN THRUST

This article introduces recent results of our research on this topic in two prospects: mining quantitative association rules and mining fuzzy association rules.

Mining Quantitative Association Rules

One method that was proposed is an adaptive numerical value discretization method that considers both value density and value distance of numerical attributes and produces better quality intervals than the two classical methods (Li et al., 1999). This method repeatedly selects a pair of adjacent intervals that have the minimum difference to merge until a given criterion is met. It requires quadratic time on the number of attribute values, because each interval initially contains only a single value, and all the intervals may be merged into one large interval containing all the values in the worst case. A method of linear-scan merging for quantizing numeric attribute values that can be implemented in linear time sequentially and linear cost in parallel was also proposed (Shen, 2001). This method takes into consideration the maximal intrainterval distance and load balancing simultaneously to improve the quality of merging. In comparison with the existing results in the same category, the algorithm achieves a linear time speedup.

Suppose that a numerical attribute has m distinct values, $I = \{x_0, x_1, ...x_{m-1}\}$, where attribute value x_i has n_i occurrences in the database (weight). Let $N = \sum_{i=0}^{m-1} n_i$ be the total number of attribute value occurrences, called instances. Without loss of generality, we further assume that $x_i < x_{i+1}$ for all $0 \le i \le m-2$ (otherwise, we can simply sort these values). Define P to be a set of maximal disjoint intervals on I, where interval $I_u \in$ P contains a sequence of attribute values $\{x_u, x_{u+1}, ..., x_{v-1}\}$ and $N_u = \sum_{i=u}^{v-1} n_i$ instances, v is the index of the next interval after I_u in P, and $0 \le u \le v \le m$. We also assume that I_u has a representative center, c_u. Initially, I_u contains only x_u, which is also its representative center.

We define the maximal intrainterval distance, denoted by $D^*(I_u; c_u)$, as follows:

$$D^*(c_u, c_v) = \max_i | x_i - c_u |$$

Assume that two adjacent intervals, $I_u = \{x_u, ...x_{v-1}\}$ and $I_v = \{x_v, ...x_{w-1}\}$, contain $N_u = \sum_{i=u}^{v-1} n_i$ and $N_v = \sum_{i=v}^{w-1} n_i$ attribute value occurrences and have representative centers $c_u = \sum_{i=u}^{v-1} x_i n_i / N_u$ and $c_v = \sum_{i=v}^{w-1} x_i n_i / N_v$, respectively, and $0 \le u \le v \le m$. The union, $I'_u = I_u \cup I_v$, of the two intervals containing $(v-u)+(w-v)=w-u$ attribute values and $N_u + N_v = \sum_{i=u}^{w-1} n_i$ instances total thus has its representative center given by the average weighted value of (c_u, n_u) and (c_v, n_v):

$$c'_u = \frac{c_u \sum_{i=u}^{v-1} n_i + c_v \sum_{i=u}^{w-1} n_i}{\sum_{i=u}^{w-1} n_i} = \frac{c_u N_u + c_v N_v}{N_u + N_v}$$

An optimal interval merge scheme produces a minimum number of intervals whose maximal intrainterval distances are each within a given threshold and whose populations are as equal as possible. Assume that the threshold for the maximal intrainterval difference is d, which can be the average interinterval differences of all the adjacent interval pairs or can be given by the system. For k intervals, let the average population (support) of each interval be $\overline{N}_k = N/k$ and the population deviation of interval I_u be $\Delta_u = | N_u - \overline{N}_k |$, where N_u is the actual population of I_u. Initially, $I_u = \{x_u\}$ and $0 \le u \le m-1$. Our strategy leads to the following algorithm for interval merging:

1. Partition $\{I_0, I_1, ..., I_{m-1}\}$ into a minimum number of intervals such that each interval has a maximal intrainterval distance not greater than d.
2. Assume that Step 1 produces k intervals: $\{I_{u_0}, I_{u_1}, ..., I_{u_{k-1}}\}$ and $0 = u_0 < u_1 ... < u_{k-1} < m-1$. For $I_{u_j} = [X_{u_j} : c_{u_j}]$, where $X_{u_j} = \{x_{u_j}, x_{u_j+1}, ..., x_{u_{j+1}-1}\}$ and c_{u_j} is the representative center of I_{u_j}, check to see if moving $x_{u_{j+1}-1}$ to $I_{u_{j+1}}$ will result in a better load balance while preserving the maximal intrainterval distance property, and do so if it will.

Noticing that $x_0 < x_1 < ... < x_{m-1}$, we can implement Step 1 simply by using a linear scan to form appropriate segments of intervals after a single pass. Starting from I_0, merge I_u with I_{u+j} for $j = 1, 2, ...$, until the next merge would result in I_u's maximal intrainterval distance greater than the threshold; continue this process until no interval to be merged remains. This process requires time $O(m)$.

Step 2 examines every adjacent pair of intervals after the merge, requiring, at most, m-1 steps. Each step checks the changes of population deviation by moving u_{j+1}-1 instances from I_{u_j} to $I_{u_{j+1}}$. That is, it considers whether the following condition holds:

$$| \Delta_{u_j} - \Delta^-_{u_j} | < | \Delta_{u_{j+1}} - \Delta^+_{u_{j+1}} |,$$

where $\Delta^-_{u_j} = | N_{u_j} - n_{u_{j+1}} - \overline{N}_k |$ and $\Delta^+_{u_j} = | N_{u_{j+1}} - n_{u_{j+1}-1} - \overline{N}_k |$.