Metarule-Guided Mining of Multi-Dimensional Association Rules Using Data Cubes

Micheline Kamber **Jiawei Han** **Jenny Y. Chiang**
Database Systems Research Laboratory, School of Computing Science
Simon Fraser University, Burnaby, BC, Canada V5A 1S6
E-mail: {kamber, han, ychiang}@cs.sfu.ca

Abstract

In this paper, we employ a novel approach to metarule-guided, multi-dimensional association rule mining which explores a *data cube* structure. We propose algorithms for metarule-guided mining: given a metarule containing p predicates, we compare mining on an n-dimensional (n-D) cube structure (where $p < n$) with mining on smaller multiple p-dimensional cubes. In addition, we propose an efficient method for precomputing the cube, which takes into account the constraints imposed by the given metarule.

Introduction

Metarule-guided mining is a interactive approach to data mining, whereby the user can probe the data under analysis by specifying hypotheses in the form of *metarules*, or pattern templates. A data mining system attempts to confirm the hypotheses by searching for patterns that match the given metarules. Metarule-guided mining increases the likelihood of finding rules that are of interest to the user and can make the discovery process more *efficient* by using the metarules to *constrain* the rule search space. For example, metarule (1) can be used to focus the data mining search towards rules disclosing which two factors combined promote the sales of *Pentium* computers.

$$\forall x \in person, P(x,y) \wedge Q(x,w) \Rightarrow buys(x, \text{"}pentium\text{"}), \quad (1)$$

where P and Q are *predicate variables*, x is a variable representing a person, and y and w are *object variables.* All variables will be instantiated to concrete values in the mining process, e.g., association rule (2) matches or *complies* with metarule (1).

$$\forall x \in person, \quad owns(x, \text{"}laptop\text{"}) \wedge income(x, \text{"}high\text{"}) \Rightarrow buys(x, \text{"}pentium\text{"}) \quad (2)$$

Metarule-guided mining is closely linked with other rule mining methods, especially association rule mining (Agrawal & Srikant 1994; Han & Fu 1995; Srikant & Agrawal 1996; Meo et al. 1996). Previous studies on data mining with metarules include (Klemettinen et al. 1994) which uses metarules as filters defining the form of interesting association rules; (Shen et al. 1996) which uses metarules to guide the mining of Bayesian data clustering rules; and (Fu & Han 1995) which uses a relation table-based structure for metarule-guided mining.

With recent progress on data warehousing and OLAP technology (Chaudhuri & Dayal 1997), it is expected that many mining tasks will be performed on data warehouses. With efficient techniques developed for computing data cubes (Harinarayan et al. 1996; Agarwal et al. 1996; Zhao et al. 1997), it is important to explore data cube-based rule mining algorithms. This motivates our study on metarule-guided mining using the data cube structure. Metarule-guided mining focuses the search on desired rule patterns, whereas data cube structures make good use of structured warehouse and precomputed aggregation information. An integration of both is likely to lead to an efficient mining method. We propose algorithms which consider: (1) when a precomputed data cube is available; and (2) dynamic construction of relevant data cubes for metarule-guided mining, otherwise. In the latter case, to mine a metarule with p distinct predicates from n relevant attributes ($p < n$), we compare construction of an n-D data cube versus construction of several smaller p-D data cubes.

Preliminaries

A metarule is a rule template of the form

$$P_1 \wedge P_2 \wedge \ldots \wedge P_m \Rightarrow Q_1 \wedge Q_2 \wedge \ldots \wedge Q_l \quad (3)$$

where P_i ($i = 1,..,m$) and Q_j ($j = 1,..,l$) are either instantiated predicates or predicate variables, and $p = m+l$ is the number of predicates in the metarule. Rule, R, complies with a metarule, M_R, iff it can be unified with M_R, e.g., (2) complies with metarule (1).

A rule $X \Rightarrow Y$, where *body* X and *head* Y each consists of a set of conjunctive predicates, is a multi-dimensional association rule iff $\{X, Y\}$ contains more than one distinct predicate. For a data set D containing tuples from a relational database, the support of a

This research is partially supported by NSERC of Canada, by NCE/IRIS, and by research grants from BC Science Council, and MPR Teltech Ltd.

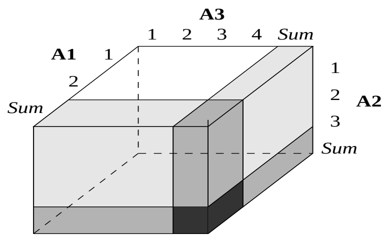

Figure 1: A 3-D data cube with summary layers

rule $X \Rightarrow Y$ in D is the probability that the tuples in D contain both X and Y. The confidence of $X \Rightarrow Y$ is the probability that a tuple contains Y given that it contains X. Typically, rules that do not satisfy user- or expert-provided *minimum support* and *minimum confidence* thresholds are considered uninteresting. A k-predicate set (i.e., containing k conjunctive predicates) is large if its support is no less than the minimum support threshold. Rule $X \Rightarrow Y$ is strong if it satisfies both minimum support and minimum confidence thresholds. A rule in which all the predicates have distinct predicate names is called a non-repetitive predicate multi-dimensional association rule. Our study of metarule-guided mining of strong, multi-dimensional association rules is confined to this case.

Example 1 Suppose a user wishes to mine a set of strong rules in a university database, associating the students' *gpa* with their *major, student_id, birth_place*, and *residence*. The task can be expressed in a data mining query language, DMQL (Han & Fu 1995), as follows.

```
mine multi-dimensional association rules
from student
in relevance to major, student_id, birth_place, residence
set template $P(s : student, x) \wedge Q(s, y) \Rightarrow gpa(s, z)$.
with min_support = 0.20, min_confidence = 0.80
```

First, the task-relevant data is extracted, i.e., the set of student records is projected onto {*gpa, major, student_id, birth_place, residence*} using an SQL-query transformed from the provided query. The minimum support threshold is used to perform dimension reduction, i.e., removing attributes with too many distinct values, such as *student_id*, and perform necessary generalization/discretization, e.g., replacing raw data values, such as 3.873 for *gpa*, by higher level concepts, such as "*excellent*". This leaves only three attributes *major, birth_place, residence* for possible matching with the predicates, P and Q. □

Our study explores rule mining using a data cube structure. An n-dimensional data cube, $C[A_1, \ldots, A_n]$, stores the group-by aggregation (count) of all possible combinations of the n attributes or *dimensions*, A_i. Each dimension contains $|A_i|$ *data rows* (where $|A_i|$ is the number of distinct values in A_i) and one *sum row*, used to store the count summation of the corresponding columns of the above data rows. A data cube can be viewed as a lattice of *cuboids*. The *n-D space* or *n-D layer* (the *base cuboid*) consists of all data cells (i.e., the count information for each n-D combination of the dimensions). The *(n-1)-D space* consists of the sum cells storing the count information for each *(n-1)*-D combination of the dimensions, and so on. The *0-D space* consists of one sum cell which stores the total number of counts (or generalized tuples) represented in the cube. A 3-D data cube is shown in Fig. 1. For sparse cubes, sparse matrix technology (Zhao et al. 1997) can be applied.

Methods for Metarule-Guided Mining Using Data Cubes

In our notation, metarule M_R is in conjunctive normal form as in Eq. (3); p is the number of instantiated predicates or predicate variables in M_R ($p \leq n$, where n is the dimension of the cube); L_k is the set of large k-predicate sets, and each member of L_k has two fields, *predicate set* and *support count*; and R is the set of all strong association rules that comply with M_R.

Mining on an existing data cube

Owing to the wide availability of data warehouses, a precomputed n-D data cube (including the summary layers) may be available for metarule-guided mining. We examine two cube-based algorithms for this case: (1) multi-D-slicing, which finds large 1-predicate sets and uses them to perform multi-dimensional slicing on the data cube. This algorithm adopts the spirit of table-based approaches to association rule mining (such as Apriori of Agrawal & Srikant 1994) as well as our earlier table-based study on metarule-guided mining (Fu & Han 1995), although here a data cube structure is used; and (2) n-D cube search, which directly examines the p-D cells of the precomputed n-D cube in order to find large p-predicate sets, L_p.

Algorithm 1 (multi-D-slicing) A multi-dimensional slicing technique for metarule-guided mining of multi-dimensional association rules.

Input: (1) Metarule, M_R, containing p distinct predicates; (2) An n-D data cube, $C[A_1, .., A_n]$, with all of its dimension space precomputed ($p \leq n$), where the cube contains the dimensions relevant to answering the specified metarule query; (3) *min_sup* and *min_conf* thresholds.

Output: R, the set of strong association rules that comply with M_R.

Method: First find the large 1-predicate sets in each dimension and then use the large predicate sets in dimension $(k-1)$ to grow large k-predicate sets by multi-dimensional slicing on the data cube until large p-predicate sets are found.

```
1) for each dimension $A_i$,
     $L_{1,(A_i)} = find\_large\_1\_predicatesets(C, A_i, min\_sup)$;
2) $L_1 = \cup_i L_{1,(A_i)}$;
3) for $(k = 2; (k \leq p)$ & $L_{k-1} \neq \emptyset; k{+}{+})$ {
4)   // intersect the (k-1) dimensions to find $L_k$, i.e.,
     $L_k = find\_large\_predicatesets(C, L_{k-1}, min\_sup)$; }
5) $R = rule\_gen(L_p, M_R, C, min\_conf)$;
6) return {$R$};
```

At each k-th iteration, only the k-D summary layer needs to be loaded into main memory. If the layer is larger than the available main memory, it can be partitioned into *chunks* (Zhao et al. 1997) and only the corresponding chunks need be loaded for efficient computation.

Rationale. Step 1 finds the large 1-predicate sets for each dimension A_i by searching the 1-D space of the cube. A *candidate* predicate set is a potentially large predicate set. If L_{k-1} is not empty (step 3), step 4 uses L_{k-1} to derive the candidate k-predicate sets by intersecting the dimension vectors in L_{k-1}, based on the principle of Apriori i.e., *every subset of a large itemset must be large* (Agrawal & Srikant 1994). The counts of the candidates in k-space are examined in order to find L_k. Search continues until L_p is found (since L_p predicate sets are needed to generate rules complying with a metarule having p predicates). In step 5, a typical rule generation method generate rules from L_p that comply with M_R and satisfy *min_conf*. □

Instead of searching for L_p from L_1, L_2, etc., the count at each p-D cell can be examined directly to find L_p, leading to,

Algorithm 2 (n-D cube search) A direct metarule-guided mining of multi-dimensional association rules by inspecting p-D cells of an *n-D* cube.

Input/Output: The same as Algorithm 1.

Method: 1. Examine the cell count of each p-D cell. If a cell count satisfies *min_sup*, then add the corresponding p-predicate set to L_p.

2. Call the same *rule_gen* procedure as in Algorithm 1, returning strong rules that comply with M_R from L_p.

Rationale. In step 1, we can find L_p directly by examining the p-D cells since the summary layers of the cube are computed. Since the entire data cube may be larger than the available memory, only the p-D summary layer need be loaded in. In step 2, given the large p-predicate sets, a typical rule generation method can be used to generate rules that comply with the metarule. □

Integration of cube construction with metarule-guided mining

When no precomputed data cube is available, the mining process must be integrated with cube computation. Metarule, M_R, contains p predicates, of which m are in the body. To mine strong rules complying with M_R, only the m- and p-D layers of the cube need be computed. We call this *abridged construction*. The p-D layer can be scanned to find L_p. The m-D layer is required to compute the confidence of rules satisfying M_R. We propose two approaches: 1) abridged n-D cube construction, which computes the p- and m-D layers of an n-D cube, while searching p-D for L_p; and 2) abridged multi-p-D cube construction, which constructs smaller, multiple p-D cubes instead of one big n-D cube for mining.

When a data cube cannot fit in main memory, the proposed algorithms employ a *cube chunking* strategy, where the cube is broken up into smaller, memory-fit chunks. Our proposed hierarchy-based chunking technique scans through the data cube in a single pass and facilitates multi-level rule mining, as well.

We describe our algorithms in two steps. First, the motivating ideas behind the two abridged cube construction algorithms are presented. For lack of space, the chunking technique is briefly discussed afterwards.

Algorithm 3 (abridged n-D cube construction) Metarule-guided mining using metarule information to construct and search only a subset of an n-D cube.

Input/Output: The same as Algorithm 1 except that the input cube contains no computed summary layers.

Method: Construct the p-D layer from the n-D data cells. Construct the m-D layer using only the p-D layer cells satisfying L_p. If the cube cannot fit into memory, use the chunking procedure described below. Call a rule generation algorithm to return strong rules.

Note. Each non-empty n-D data cell is visited only once, with its count accumulated in each of the corresponding p-D planes. The m-D layer is computed from a scan of the p-D cells satisfying *min_sup*. A flag can be set to indicate that an m-D plane has been constructed, so that none of the counts on a p-D plane is inappropriately added more than once to the same m-D plane. □

Instead of constructing an n-D cube for mining a p-predicate metarule (where n is the number of dimensions relevant to the mining task), the mining can be performed on smaller p-D cubes ($p < n$).

Algorithm 4 (abridged multi-p-D cube construction) Metarule-guided mining of a p-predicate metarule with n related dimensions by construction of $\binom{n}{p}$ p-D cubes.

Input/Output: The same as Algorithm 1 except that the input cube contains no computed summary layers.

Method: Since there are $\binom{n}{p}$ ways to choose p dimensions, compute $\binom{n}{p}$ p-D cubes without summary layers, then compute the m-D summary layer from the "large" p-D data cells. A chunking algorithm is used, as described below, to compute the relevant layers if there is insufficient memory. Since the (p-D) data cells already have their counts associated with them, only the m-D layer of each cube is computed. □

Computing summary layers by chunking

When there is insufficient main memory to hold a data cube for computation, *chunking* techniques which partition the data cube into small subcubes ("chunks") can be used (Sarawagi & Stonebraker 1994; Agarwal et al. 1996; Zhao et al. 1997).

An efficient, array-based algorithm for simultaneous multidimensional aggregation has been studied recently (Zhao et al. 1997). At any given time, at least one chunk is loaded into memory. The data cells of the chunk are scanned, and the corresponding summary layers of the data cube are updated. This requires memory for the chunk being processed, and for the portions of the summary layers being updated. If the memory is sufficient to hold the chunk and summary layer portions, the summary layers can be computed in a single sweep through each cube chunk. Otherwise, some chunks must be scanned more than once.

A similar method for cube computation has been proposed in our study. The method is based on "chunking" a cube according to dimension hierarchies, computing each chunk as a subcube *with summary layers*, and merging summary layers of subcubes into a cube summary layer. Since each chunk is partitioned according to dimension hierarchies, a chunk forms a semantically meaningful entity and its computation is equivalent to computing an interesting, meaningful subcube which may facilitate mining of multiple-level rules (Han & Fu 1995). Each chunk and its summary layers are small enough to fit in main memory. One scan of a chunk derives all of its summary layers, and these layers are sufficient for computing the summary layers of the entire cube. Thus, a single scan of each chunk is sufficient for computing the entire cube.

Although each chunk is accessed only once, the subsequent computation of cube summary layers may fetch the summary layers of multiple chunks, costing I/O operations. However, the ordering in which summary layers are computed can be explored, e.g., by computing some summary layers simultaneously if there is sufficient memory. Therefore, this hierarchy-based chunking and prior computation of chunk summary layers is an interesting alternative to (Zhao et al. 1997) for cube computation. This is especially beneficial for metarule-guided mining since only a few cube layers (i.e., p and m) need to be computed.

Performance Study

Based on our performance study (Kamber et al. 1997), we recommend the n-D cube search algorithm when the data cube is available. Regardless of the number of generalized tuples, n-D cube search must examine the same number of cells (i.e., the p-D layer) in order to find L_p. Hence, it exhibits better scalability than multi-D-slicing, whose execution time is dependent on the number of candidates found, which typically increases with the data set size. When no data cube is available, we recommend abridged multi-p-D cube construction over abridged n-D cube construction since the former exhibits greater scalability as the number of relevant dimensions (n) and values per dimension grow.

Discussion and Conclusions

Previous methods for metarule-guided mining of association rules have primarily used a table-based structure, requiring costly, multiple scans of the data. We proposed four algorithms which explore a data cube structure for metarule-guided mining of multi-dimensional association rules. When an appropriate n-D cube is available, an n-D cube search of the p-D layer cells can be applied (where p is the number of metarule predicates, $p \leq n$). Otherwise, information from the metarule, such as the number of predicates, can be used for efficient construction of a relevant cube and mining of the required rules. Preliminary results show that when the number of relevant dimensions or values per dimension is large, it is more efficient to construct and mine from multiple, smaller p-D cubes than from one large n-D cube. For efficient cube construction, a hierarchy-based chunking algorithm was proposed which requires the scan of each chunk at most once and may facilitate mining of multiple-level rules. This study is confined to mining rules with no repetitive predicates. Efficient methods for metarule-guided mining of more general forms of rules using data cubes is an interesting topic for future research.

References

Agarwal, S.; Agrawal, R.; Deshpande, P. M.; Gupta, A.; Naughton, J. F.; Ramakrishnan, R.; and Sarawagi, S. 1996. On the computation of multidimensional aggregates. In *Proc. VLDB'96*, 506–521.

Agrawal, R., and Srikant, R. 1994. Fast algorithms for mining association rules. In *Proc. VLDB'94*, 487–499.

Chaudhuri, S., and Dayal, U. 1997. An overview of data warehousing and OLAP technlogy. *SIGMOD Record* 26:65–74.

Fu, Y., and Han, J. 1995. Meta-rule-guided mining of association rules in relational databases. In *Proc. Int'l Workshop on Integration of Knowledge Discovery with Deductive and Object-Oriented Databases*, 39–46.

Han, J., and Fu, Y. 1995. Discovery of multiple-level association rules from large databases. In *Proc. VLDB'95*, 420–431.

Harinarayan, V.; Rajaraman, A.; and Ullman, J. D. 1996. Implementing data cubes efficiently. In *Proc. SIGMOD'96*, 205–216.

Kamber, M.; Han, J.; and Chiang, J.Y. 1997. Using data cubes for metarule-guided mining of multi-dimensional association rules. CS-TR 97-10, Simon Fraser Univ, http://db.cs.sfu.ca/sections/publication/kdd/kdd.html.

Klemettinen, M.; Mannila, H.; Ronkainen, P.; Toivonen, H.; and Verkamo, A. 1994. Finding interesting rules from large sets of discovered association rules. In *Proc. CIKM'94*, 401–408.

Meo, R.; Psaila, G.; and Ceri, S. 1996. A new SQL-like operator for mining association rules. In *Proc. VLDB'96*, 122–133.

Shen, W.; Ong, K.; Mitbander, B.; and Zaniolo, C. 1996. Metaqueries for data mining. In Fayyad, U.; et al. (eds), *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press. 375–398.

Sarawagi, S.; and Stonebraker, M. 1994. Efficient organization of large multidimensional arrays. In *Proc. ICDE'94*, 328-336.

Srikant, R.; and Agrawal, R. 1996. Mining quantitative association rules in large relational tables. In *Proc. SIGMOD'96*, 1–12.

Zhao, Y.; Deshpande, P. M.; and Naughton, J. F. 1997. An array-based algorithm for simultaneous multdimensional aggregates. In *Proc. SIGMOD'97*, 159–170.