An Efficient Algorithm for Hierarchical Online Mining of Association Rules

J.Ashok
Professor and Head, Department of Information Technology, GCET, Hyderabad, India.
Email: jammiashok@yahoo.co.in
P.Satish
Asst.Professor, Department of Information Technology, VIET, Hyderabad, India.
Email: satish.puli@yahoo.co.in
Y.Raju
Assoc.Professor, Department of Information Technology, GCET, Hyderabad, India.
Email:raju.yeligeti@gmail.com

ABSTRACT

Several multi-pass algorithms have been proposed for Association Rule Mining from static repositories. However, such algorithms are incapable of online processing of transaction streams. In this paper we introduce an efficient single-pass algorithm for mining association rules, given a hierarchical classification amongst items. Processing efficiency is achieved by utilizing two optimizations, *hierarchy aware counting* and *transaction reduction*, which become possible in the context of hierarchical classification. We also propose a modified algorithm for the rule generation phase which avoids the construction of an explicit adjacency lattice.

Date of Submission: April 09, 2010 Date of Acceptance: June 09, 2010

1. Introduction

The aim of Association Rule Mining is to find latent associations among data entities in database repositories, a typical example of which is the transaction database maintained by a supermarket. An association rule is an implication of the form A=> *B*, which conveys that customers buying set of items *A* would also with a high probability buy set of items *B*. The concept of association rule mining was first introduced in [1]. Typically the problem is decomposed into two phases. Phase I of the problem involves in finding the frequent item sets in the database, based on a pre- defined frequency threshold *minsupport*. Phase II of the problem involves generating the association rules from the frequent item sets found in Phase I. Typically, the reported approaches in Phase I re- quire multiple passes over the transaction database to determine the frequent item sets of deferent lengths [1, 2, 3]. All these approaches assume that a static database is available, so that multiple scans can be made over it. With online systems, it is desirable to make decisions on the y, processing data-streams in- stead of stored databases. Some work has been re- ported which is related to online versions of the rule- mining algorithm [4, 5]. However, in these reports, reference is still made to static repositories. In this paper, we aim at a true online algorithm, capable of processing online streams of transactions. Assume that the algorithm has computed its result up to and including the first *n* transactions. A true online algorithm should be capable of updating the result for the (*n* + 1) transaction, without requiring a re-scan over the past *n* transactions. In this way such an algorithm can handle transaction streams. In fact it is true that items in an online shopping mart or a supermarket are categorized into sub-classes, which in turn make up classes at a higher level, and so on. Besides the usual rules that involve individual items, learning association rules at a particular sub- class or class level is also of much potential use and significance, e.g. an item-specific rule such as customers buying Brand A sports shoes tend to buy Brand B tee- shirts" may be of less practical use than a more general rule such as Customers buying sports shoes tend to buy tee-shirts". With this aim, use can be made of commonly employed hierarchical classification of items to devise a simple and efficient rule mining algorithm. [6] Proposes a single-pass algorithm for hierarchical online association rule mining; in this paper, we refer to this algorithm as HORM. The present work carries forward the idea of [6], and proposes an efficient algorithm for Phase I. The present work also looks at Phase II, i.e. the generation of association rules. [5] Proposes an algorithm to generate non-redundant rules. We present a modified algorithm for Phase II that better suits the need to mine hierarchical association rules.

2. Theories

2.1 Basic Concepts and Problem Formulation

Hierarchical classification of data means that the items which make up a transaction are categorized into Classes, sub-classes, and so on. Evidently it is possible to mine for two types of rules: an item-specific rule such as Customers buying soap of brand A tend to buy canned soup of brand B", or a more general Rule such as Customers buying soaps tend buy canned soup". The latter is an association on classes or sub-classes, rather than on individual items. Let *I* be the set of all items stored in, say, a typical supermarket. We suppose that, at each level of classification, a fixed number *M* of classes, sub-classes or items are present. At the root level we have classes *C*1*; C2; C3*: *CM*. At the next level, for a class *Ck*, we will have the *M* sub-classes *Ck*1*;Ck*2 *: : :CkM*. For *jIj* = 20000, and with *M* = 12, for example, we will need four levels of classification; the last level will contain individual items stored in transaction, which will be coded as *Cjklm*, i.e. one index for each level of classification. A hierarchical association rule is an implication of the type *X) Y* where *X; Y* are disjoint subsets of the sub-classes of some *C*®, the parent class of *X* and *Y* . As usual, support for association rule *X / Y* is defined as the fraction of transactions in the transaction database which contain *X [Y* ; confidence of the rule is defined as the fraction of transactions containing *X* which also contain *Y* . We denote the support and confidence of rule *X) Y* as *supp*(*X) Y*) and *conf*(*X) Y*) respectively. We may also write *XY* to represent *X => Y*. Subsets of sub-classes of class *C* are elements of the power-set of the set of sub-classes of *C*. For a given class *C*, the counts of all subsets occurring in the transaction database are stored in an integer array, called count array, of size 2*M*. The natural bitmap representation of a subset can be used directly as the index of the corresponding cell in the count array [6]. Note that these bitmaps also form the so-called *adjacency lattice[7]*, which are used in Phase II to generate non-redundant rules. *M* denotes the number of sub-classes of each class in the hierarchical classification tree; we may refer to *M* as the *degree of classification*. *C* denotes the height of the classification tree, assumed to be regular with degree *M*. Individual items make up leaf nodes of the tree, and the path from the root to a leaf node makes up the code of the corresponding item. Therefore *C* is also the item code length of the classification scheme employed. The string *X*1*X*2 *: : :XC* denotes a particular class, sub-class or item code, where each *Xi* is either the asterisk ¤, or a classification symbol denoting a class or sub-class at level *i*. The notation scheme used is the following: 1. The root node of the classification tree is coded by a string of asterisks of length *C*. 2. An item (leaf node in the classification tree) is coded as *X*1*X*2 *: : :XC*, where *8i;Xi 6*= ¤. 3. An intermediate class or sub-class in the tree at level *i* is coded as *X*1 *: : :Xi* ¤*i*+1 *: : :* ¤*C* , i.e. *I* classification symbols followed by *C* ¡ *i* asterisks.

2.1.1 Horn Algorithm

From the classes and sub-classes which make up the classification tree, the user selects a *set of classes of interest* [6], to be denoted here as *SIC*. Association rules to be mined are of the type *X => Y* where *X* and *Y* are disjoint subsets of a class or sub-class of interest. The problem of hierarchical association rule mining is now defined as: Find all association rules of the type *X => Y*, within each class of interest in *SIC*, which have a specified minimum support and confidence in the transaction database or stream. To find associations within a class or sub-class of interest, we need to maintain counts for all its subsets. For the class *A*¤¤¤, with *M* = 4, for example, we need to count the occurrences in the transaction database of all the subsets of *fA*1¤¤*;A*2¤¤*;A*3¤¤*;A*4¤¤*g*. Clearly there are 2*M* ¡ 1 non-empty subset combinations for a sub-class with *M* elements. Therefore a count array of size 2*M* ¡1 needs to be maintained for each class or sub-class of interest. HORM algorithm takes as input the transaction database *D*, and a set of classes or sub-classes of interest, denoted *SIC*. After one scan over the database, the count-arrays of all classes or sub-classes of interest contain the number of occurrences, i.e. support values, in the transaction database of all the subsets of the classes or sub-classes of interest. The time complexity of this algorithm is *O*(*jDjK*2*M*) [6]. The memory requirement of HORM is *K*2*M*, since each element of *SIC* requires an array of size 2*M*.

2.2 Enhancements Proposed

2.2.1 Hierarchy-aware Counting

In HORM, each transaction is checked against all the classes or sub-classes in *SIC*. But suppose we have two classes or sub-classes in *SIC* of which one is itself a sub-class of the other. In HORM, the per-transaction code is executed once for each of these elements of *SIC*, without taking into account this hierarchical relationship between the two. But clearly if the first iteration suggests that the current transaction does not support, say *PQ*¤¤, we do not need to iterate for any of its sub-class such as *PQR*¤. We apply this intuition to speed up the algorithm. If a transaction does not support a class or sub-class, it does not support any of its sub-classes either. We call this first enhancement *hierarchy-aware counting*.

2.2.2 Transaction Reduction

This second enhancement reduces the computation within the inner loop. For every class or sub-class in *SIC*, HORM processes the current transaction in its entirety. However, suppose we have two classes or sub-classes in *SIC* which do not share an ancestor- descendant relationship. Once we have matched the entire transaction against the first class or sub-class clearly it is not necessary to again match the entire transaction against the second one as well. Suppose *A* and *B* are two classes of interest, and let the current transaction *T* be *fA*1*Q*6*;A*2*P*6*;B*2*Q*6*;B*1*Q*7*;A*2*P*7*;B*2*P*7*g*. While *T* is being checked against *A*¤¤¤, the algorithm in fact traverses through the items of *T* and finds the sub- transaction *T*=*A*¤¤¤ = *fA*1*Q*6*;A*2*P*6*;A*2*P*7*g*, which may be called the *projection* of class *A*¤¤¤ on *T*. Clearly *T*=*A*¤¤¤ does not contain any items that belong to *B*¤¤¤, because the sub-classes of *A*¤¤¤ and *B*¤¤¤ are disjoint. Thus we can remove *T*=*A*¤¤¤ from *T* and pass the remaining items *T*1 = *T* ¡ *T*=*A*¤¤¤ to match against *B*¤¤¤. Thus the part of a transaction that is a projection of a class can be removed to obtain a reduced transaction to match against disjoint classes. We call this second enhancement *transaction reduction*.

2.2.3 Non -Redundant Rule Generation

The version implemented in the present work is based on the basic concepts proposed in [5]. The hierarchical rule mining technique described here does not require a separate adjacency lattice of the classes or subclasses of interest. The count arrays described above can themselves be viewed as adjacency lattices used in [5], leading to very clean design and implementation.

3. Experimental Results

We have evaluated the performances of HORM and EHORM, as well as that of the Phase II sub-problem. All the experiments are conducted using synthetic datasets generated using the process outlined below.

The synthetic data generator described in [2] provides the basic model of the data generator used in the present work. Since we are working with hierarchically classified items rather than unclassified items, the necessary enhancements are made to the dataset generation algorithm, with an additional parameter being the classification degree *M*. Integer item codes are converted to hierarchical item codes by the taxonomy determined by *M*. Parameters for the synthetic hierarchical data generator are listed in Table 1.

Figure 1 shows the performance of HORM and EHORM with varying *M*, the classification degree. Other parameters are fixed at *T* = 5, *I* = 5, *D* = 100*k*, *N* = 1000, and *jLj* = 500. Execution times are seen to increase with increasing *M*, due to the fact that the *Parameter Description T* Average transaction size *I* Average size of frequent class-set *D* Number of transactions in database *M* Classification degree *L* Number of frequent class-sets *N* Number of items.

Parameter	**Description**
T	Average transaction size.
I	Average size of frequent class-set.
D	Number of ransactions in database.
M	Classification number
L	Number of frequent class sets.
N	Number of items

Table1: Parameters of the synthetic data generator

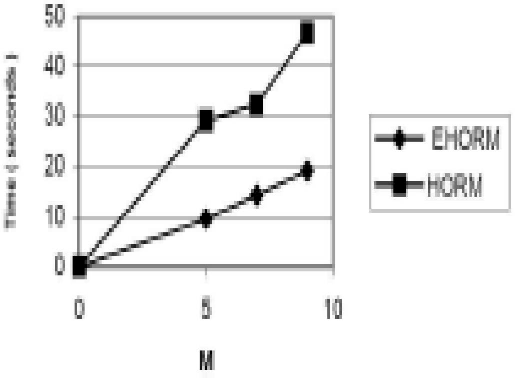

Figure 1: Runtime as a function of *M*

Classification tree grows horizontally with increasing *M*, reducing its height but increasing the number of iterations while accessing the count array. Figure 2 shows the variation of execution times of EHORM and HORM with varying *T*, the average transaction size, with the other parameters fixed at *I* = 5, *D* = 100*k*, *N* = 2000, *juju* = 500, and *M* = 6. Figure 3 shows the execution times with varying *I*, the average frequent class-set size. The number of association rules generated, while varying the confidence threshold *mincon f*, has been collected for three types of datasets. The aggregate obtained results are shown in Figure 4. As expected, the number of rules generated is seen to decrease with increasing *mincon f*, as that value is varied from 0*:*6 to 1*:*0. The redundancy analysis is carried out by considering the redundancy ratio as the ratio of total rules that could have been generated, without the redundancy check being applied, to the essential or non-redundant rules finally generated.

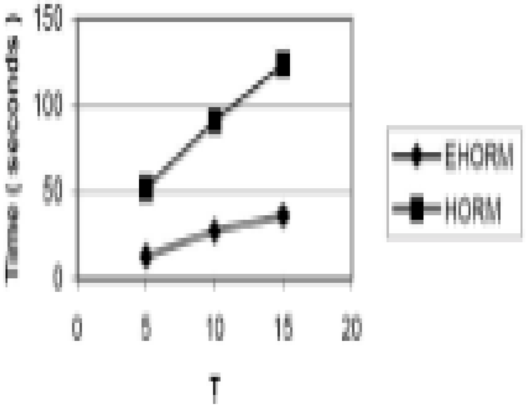

Figure 2: Runtime as a function of *T*

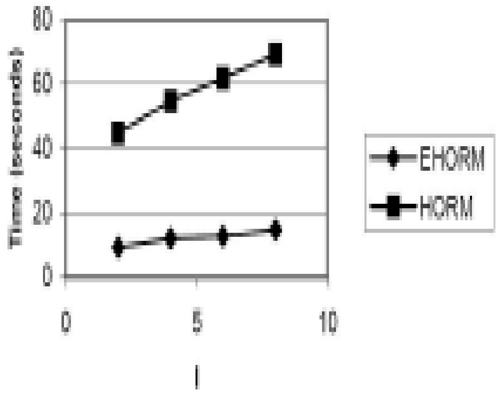

Figure 3: Runtime as a function of *I*

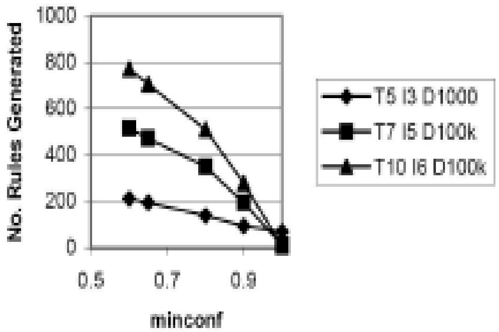

Figure 4: Number of rules generated as a function of *mincon f*

Experiments were carried out to obtain the redundancy ratio as a function of *mincon f*, for three datasets. From the results, shown in Figure 5, it is seen that the redundancy removed from the mined hierarchical association rules is of the same order as that reported in [5].

4. Conclusions and Discussion

We have proposed a new algorithm, an efficient hierarchical online rule mining, or EHORM, which optimizes the time requirements of the earlier reported algorithm HORM [6]. The proposed new algorithm incorporates two specific enhancements: *hierarchy-aware counting* and *transaction reduction* in the Phase I sub-problem. For Phase II, we have modified in a natural way the algorithm of [5] to model the generation of hierarchical association rules. Some anomalies related to the use of *confidence* in selecting rules have been discussed in [3], and the more intuitive concept of *conviction* is proposed as an alternative[9].

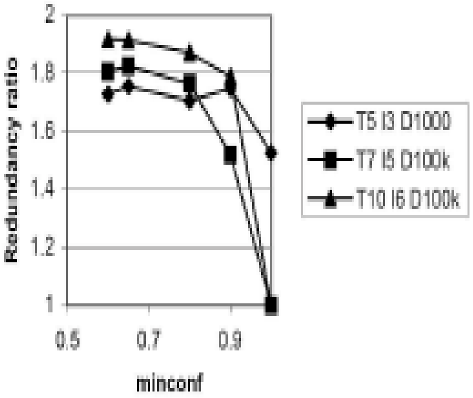

Figure 5: Redundancy ratio as a function of *mincon f*

The natural question that arises then is: How should the Phase II algorithm described here be modified to work with *conviction* in place of *confidence* as the criterion for the selection of rules? Within the present model, the generation of cross class rules, of the type *XY) Z* where *X*, *Y* and *Z* may belong to two deferent classes or sub-classes, will be of much practical use. Online generation of rules can be further enhanced by providing user-selectable parameters *mincon f* and *minsupport*, allowing the algorithm to respond and adapt to user requirements and feedback. Several authors, for example [3, 4], have argued in fact that user input is an important characteristic of any data-mining algorithm[10].

5. References

[1] R. Agrawal, T. Imielinski and A. Swami, Mining Association Rules between Sets of Items in Large Databases, Proceedings of the ACM SIGMOD Conference, 1993.

[2] R. Agrawal and R. Srikanth, Fast Algorithms for Mining Association Rules, Proceedings of the 20th VLDB Conference, 1994.

[3] S. Brin, R. Motwani, J. D. Ullman and S. Tsur, Dynamic Itemset Counting and Implication Rules for Market Basket Data, Proceedings of the ACM SIGMOD International Conference on Management of Data, 1997.

[4] C. C. Aggarwal and P. S. Yu, Online Generation of Association Rules, Proceedings of the Four- teenth International Conference on Data Engineering, ICDE, 1998.

[5] C. C. Aggarwal and P. S. Yu, A New Approach to Online Generation of Association Rules, IEEE Transactions on Knowledge and Data Engineering, Vol.13, No. 4, July/August 2001.

[6] N. Jotwani, Hierarchical Online Mining for Associative Rules, Proceedings of the 12th International Conference on Management of Data; COMAD 2005b, 2005.

[7] B. Mozafari, H. Thakkar, and C. Zaniolo.
Verifying and mining frequent patterns from large windows over data streams. *ICDE*, pages 179-1882008M. Ester, H.-P. Kriegel, J. Sander, M. Wimmer, and X. Xu. Incremental clustering for mining in a data warehousing environment. In *VLDB*, pages 323-333, 1998.

[8] F.Guillet and H. J. Hamilton, editors. *QualityMeasures in Data Mining*. Studies in Computationa Intelligence. Springer, 2007

[9] **Datta S, Datta S:** Methods for evaluating clustering algorithms for gene expression data using a reference set of functional classes. *Bioinformatics* 2007, 23:3113-8. *Bioinformatics* 2007, 23:3113-8.

[10] *BMC* Davies MN, Secker A, Freitas AA, Mendao M, Timmis J, Flower DR: On the hierarchical classification of G protein-coupled receptors. *Bioinformatics* 2006, 7:397

Authors Biography

Prof J. Ashok is currently working as Professor and Head of Information Technology at Geethanjali College of Engg. & Technology, Hyderabad, A.P, INDIA. He has received his B.E. Degree from Electronics and Communication Engineering from Osmania University and M.E. with specialization in Computer Technology from SRTMU, Nanded, INDIA. His main research interest includes neural networks, data retrieval process, Data Mining and Warehousing and Artificial Intelligence. He has been involved in the organization of a number of conferences and workshops. He has been published number of papers in national and International journals and conferences. He is currently doing his Ph.D from Anna University.