Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol I
IMECS 2009, March 18 - 20, 2009, Hong Kong

Use of Domain Knowledge for Fast Mining of Association Rules

Dr. Mahesh Motwani Dr. J.L. Rana Dr R.C Jain

***Abstract* - Data Mining is often considered as a process of automatic discovery of new knowledge from large databases. However the role of the human within the discovery process is essential. Domain knowledge consists of information about the data that is made available by the domain experts. Such knowledge constrains the search space and enhances the performance of the mining process. We have developed an algorithm that makes use of domain knowledge for efficient mining of association rules from university course enrollment database. The experimental results show that the developed algorithm results in faster mining of association rules from the elective course university dataset as compared to mining the same patterns with an association rule-mining algorithm that does not makes use of domain knowledge.**

***Index Terms* - Data Mining, Association Rules, Large itemsets, Domain Knowledge.**

I. INTRODUCTION

Association rule mining is an important data-mining task that aims to extract interesting correlations, frequent patterns or associations among sets of items in the transaction databases or other data repositories. To discover interesting relationships among attributes from large volumes of data efficient algorithms are needed. Thus an essential research issue surrounding association rule mining is to find fast and effective association rule mining algorithms.

Domain knowledge consists of information about the data that is made available either through some other discovery process or from a domain expert. Using this background information i.e. domain knowledge will constrain the search space and the rule space, thus enhancing the performance of the mining process. In this paper we show how domain knowledge is useful for data mining. The paper is organized as follows. In Section 2 we describe association rule mining and *apriori* algorithm for mining of association rules. In section 3, we describe the use of domain knowledge in data mining. Section 4 describes the relevance of association rule mining in university course enrollment dataset. In section 5 we develop a new algorithm for efficient mining of association rules from university course enrollment dataset in presence of domain knowledge. In section 6 we show the experimental results. Section 7 contains concluding note and finally section 8 contains references.

Manuscript received July 2, 2008.
Dr. Mahesh Motwani is Reader in Computer Science & Engineering department with Government Engineering College, Jabalpur (M.P. – India). (Phone 91-0755-2665500, Fax 91-0761-2431355, email mahesh_9@hotmail.com)
Dr. J.L. Rana is Professor Computer Science & Engineering at MANIT Bhopal (M.P. – India). (Phone 91-0755-2428841, 9303131297, email jl_rana@yahoo.com)
Dr R.C Jain is Director at Samrat Ashok Technological Institute Vidisha (M.P. – India). (Phone 91-07592-251045, Fax 91-07592- 250124, email dr.jain.rc@gmail.com)

II. ASSOCIATION RULE MINING

Discovery of association rules is an important class of data mining and aims to discover interesting relationships among attributes in the data. Formally, as defined in [1], the problem of mining association rules is stated as follows: Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of literals, called items. Let D be a set of transactions, where each transaction T is a set of items such that $T \subseteq I$. Associated with each transaction is a unique identifier, called its transaction id *TID*. A transaction T contains X, a set of some items in I, if $X \subseteq T$. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$ and $X \cap Y = \Phi$. The meaning of such a rule is that transactions in D, which contain the items in X, tend to also contain the items in Y. The rule $X \Rightarrow Y$ holds in D with confidence c if c% of transactions in D that contain X also contain Y. The rule $X \Rightarrow Y$ has support s in the transaction set D if s% of transactions in D contains $X \cup Y$. The problem of mining association rules is to generate all association rules that have support and confidence greater than the user-specified minimum support and minimum confidence respectively.

The problem of mining association rules is decomposed into two sub problems:

1. Finding large itemsets: In this sub problem, the aim is to find all sets of items (*itemsets*) that have transaction support above a user specified minimum support. The support for an itemset is the number of transactions that contain the itemset. Itemsets with minimum support are called *large* itemsets. Apriori algorithm for finding all large itemsets is described in section 2.1.

2. The large itemsets are then used to generate the desired rules. For every large itemset l, all non-empty subsets of l are derived. For every such subset a, an association rule of the form $a \Rightarrow (l - a)$ is obtained if the ratio of support (l) to support (a) is at least equal to minimum confidence.

A number of algorithms for association rule mining have been proposed in the literature [1], [2], [8]-[19].

A. Apriori Algorithm

Apriori [2] is the very first efficient algorithm to mine association rules. It works iteratively and makes as many passes over the database as the length of maximal itemset. An itemset is maximal large if it has no superset that is large. Let an itemset having k items be denoted as k-

itemset. The first pass of the algorithm simply counts item occurrences to determine the large 1-itemsets. A subsequent pass, say pass *k*, consists of two phases. First, the set of large {*k*-1} itemsets L_{k_1} found in the (*k*-1) $_{th}$ pass are used to generate the set of candidate *k*-itemsets C_k, using the apriori-gen function. Next, the database is scanned and the support of candidates in C_k is counted using the counting based method in order to determine the large *k*-itemsets L_k.

The *apriori-gen* function takes as argument L_{k_1}, the set of all large (*k* - 1)-itemsets. It returns a superset of the set of all large *k*-itemsets. The function works as follows:

1. The first step is *join* in which L_{k_1} is joined with L_{k_1}. Select two itemsets *p*, *q* from L_{k-1} such that first *k*-2 items of *p* and *q* are same, and then form a new candidate *k*-itemset *c* as: Common *k*-2 items + 2 differing items

2. Next in the *prune* step, prune those *c*, such that some (*k*-1) subset of *c* is not in L_{k-1} this is because all subsets of a large itemset must also be large.

III. DOMAIN KNOWLEDGE IN DATA MINING

Various researchers [3]-[7] have paid attention on the role of domain knowledge in data mining. Domain knowledge consists of information that is not explicitly presented in the database rather it is made available from a domain expert. The huge size of data forces the use of techniques for focusing on specific portions of data, which requires some additional information in the form of background knowledge, about the form of data and constraints on it. The advantage of background information, i.e. domain knowledge, is that it constrains the search space and the rule space, thus enhancing the performance of the mining process. For example, in a personnel database the domain expert may provide the following domain knowledge: *If* (*salary* > 50,000) then *post* = {Director, Principal, Professor}. This rule specifies that the post of anyone with salary more than Rs. 50,000 per month is a Director, Principal or Professor. It may be possible that there are people with other post whose salary is more than Rs. 50,000 but they are not of interest, and should be pruned off. Thus this rule specifies a subset of domain of the attribute *post* that is of interest whenever the antecedent of the rule is true.

Use of domain knowledge reduces the time experts have to spend on identifying and interpreting interesting findings. When a set of rules is generated from the dataset, the pre-conceived knowledge about the domain can help the user to determine how well these rules match or deny users existing knowledge. For example, if from the pre-conceived knowledge user believes that an old person (age greater than 60 years) is likely to have blood pressure. This belief can be expressed as a rule: *Age > 60* $\Rightarrow$ *Blood-Pressure = Yes* but if the rule generated from the dataset is: *Age >60* $\Rightarrow$ *Blood-Pressure = No,* then this rule is contrary to the user's knowledge and is interesting. Thus if a newly discovered rule is surprising to the user as it is unexpected, then it is certainly interesting.

Another form in which domain knowledge is used is, generalizing the attribute values with the domain knowledge available, e.g. a *date* field can be generalized into decades like {twenties, thirties etc.}, *Age* field can be generalized into age brackets like {child, Adolescent, young, middle-aged, old, very old}. Generalizing the attributes result in the advantage that patterns mined from the databases are more meaningful and also more useful for decision making. For example a rule: *if a person is born on 7th May 1969, he buys and drives Maruti Alto car* is less useful to marketing division of company than the rule*: if a person is in his late thirties, he buys and drives Maruti Alto car*

In a retail store, the products that generate good profits should be retained and poor profit items are removed. Hence, it is interesting to find a subset of products to be discontinued so that the profit can be maximized. It may be simple to sort items by their profits and do the selection, but it is important to consider the factor of cross selling because there can be items that do not generate much profit by themselves but cause the sale of other profitable items. This is because the customers may not make purchase of high profit items from a store, if they know they cannot get some of the other items in that store. Not considering the cross selling effects may result in loss to storekeeper. Association rules are used to model the cross selling effects among the items. The cross selling factor may be determined from the domain knowledge of the experts. This problem of optimal product selection with association rules and cross selling effects is considered in [7].

IV. MINING ASSOCIATION RULES FROM UNIVERSITY COURSE ENROLLMENT DATA

University course enrollment database contains registration information of students with one tuple per student containing the courses that a student has registered for. When there is provision for students to select electives of their choice during each semester, Patterns like combination of electives registered by different students may be useful. Each itemset consists of elective codes. The support of an itemset *I* is the number of students whose list of electives taken include all the electives in *I*. If *minsup* is the minimum support, then *I* is large if at least *minsup* number of students have taken all the electives included in *I.*

Associations of the form $X \Rightarrow Y$, where both *X* and *Y* are subsets of electives can be discovered from the course enrollment database in order to discover relationships between elective subjects taken by the students. The information from the rules discovered can be beneficial for the university board of studies for curriculum development etc. In section 5 an efficient algorithm is given for discovering such association rules from elective courses enrollment database with use of domain knowledge.

V. ALGORITHM FOR FAST ASSOCIATION MINING USING DOMAIN KNOWLEDGE

Table I shows the list of subjects in each elective and the corresponding elective code given for each subject.

Consider the course enrollment database of Table II, containing list of electives chosen by different students. A student can chose only one subject from the list of various subjects available in a particular elective group. This prior domain knowledge restricts the search space resulting in fast mining of association rules as discussed below.

Each elective field is partitioned into intervals with interval size equal to the number subjects in that particular elective. Each elective field of Table II is then mapped into intervals shown in Table I. The result is shown in Table III.

Based on the domain knowledge available that, a student can chose only one subject from the list of subjects available in any particular elective, it is obvious that support of any candidate *k*-itemset, with all the *k* items (subjects) belonging to the same elective is zero, and cannot be large. So there is no need to generate candidate itemsets containing subjects from the same elective.

The following algorithm has been developed for mining association rules making use of this domain knowledge. The algorithm is given in Fig. 1:

```
INPUT: Dataset as shown in Table III.
       Number of elective groups: $n$

  1. Generate large 1-items $L_1$ by scanning the dataset.
  2. Divide $L_1$ into groups $R_i$ ($i$ = 1 to $n$) such that $R_i$ contains the elective codes belonging to $i^{th}$ elective group.
  3. Candidate 2-itemsets $C_2$ are generated from
     $R_1 (R_2+\ldots+R_n) + R_2 (R_3+\ldots+R_n) +\ldots+R_{n-2} (R_{n-1}+R_n) +R_{n-1} (R_n)$

     From the candidate 2-itemsets, we find the large 2-itemsets $L_2$ using counting based method of apriori algorithm.
  4. Candidate 3-itemsets $C_3$ are generated from $L_2$ as follows
     $R_1R_2 (R_1R_3+\ldots+R_1R_n) + R_1R_3 (R_1R_4+\ldots+R_1R_n) + \ldots+$
     $R_2R_3 (R_2R_4+\ldots+R_2R_n) + R_2R_4 (R_2R_5+\ldots+R_2R_n) + \ldots + R_{n-2}R_{n-1} (R_{n-2}R_n)$
     From the candidate 3-itemsets, we find
     the large 3-itemsets $L_3$ using subset counting based method of apriori algorithm.
  5. For $k \geq 4$, till the candidate $k$-itemset is empty.
     a) Generate candidate $k$- itemsets from large ($k$-1) itemsets using apriori-gen
     b) Find large $k$-itemsets from candidate $k$-itemsets using subset counting based method of Apriori algorithm.
  6. Derive association rules from the large itemsets found.
```

Fig. 1: Algorithm for mining association rules using domain knowledge

In the example database of Table III, assuming *minsup* = 2 transactions, we get L_1= E11, E12, E13, E21, E22, E23, E32, E33, E42, E43.

The *Apriori* algorithm will generate 10*9/2 = 45 candidate 2-itemsets.

The developed algorithm gives R_1= {E11, E12, E13}, R_2= {E21, E22, E23}, R_3 = {E32, E33} and R_4 = {E42, E43}. This algorithm generates all the candidates generated by *Apriori* except the following candidate 2-itemsets, as their count is always zero:

E11E12, E11E13, E12E13, E21E22, E21E23, E22E23, E32E33, E42E43.

Thus only the remaining 37 candidate 2-itemsets are generated. This pruning of some candidate 2-itemsets makes our algorithm faster than the *apriori* algorithm. From these 37 candidate 2-itemsets, the large 2-itemsets L_2 are then found by scanning the database.

The number of candidate-2 itemsets generated with the developed algorithm is much less than those generated using *Apriori* based algorithms, which generate $|L_1| (|L_1| - 1)/2$ number of candidate 2-itemsets from large items using the *apriori_gen* function.

Table I: List of Electives

Elective1	Network Management (E11)	Advance Computer Architecture (E12)	Software Project Management (E13)
Elective2	Network Security (E21)	CAD of Digital Systems (E22)	Fuzzy Logic (E23)
Elective3	ATM Networks (E31)	Embedded Systems (E32)	Digital-Image Processing (E33)
Elective4	Wireless Networks (E41)	Data Mining & Data Warehousing (E42)	Pattern Recognition (E43)

Table II: Elective enrollment database

Enrollment number	Elective1	Elective2	Elective3	Electiv4
1	Network Management	Network Security	Digital-Image Processing	Pattern Recognition
2	Advance Computer Architecture	Fuzzy Logic	Embedded Systems	Data Mining & Warehousing
3	Software Project Management	Network Security	ATM Networks	Data Mining & Data Warehousing
4	Network Management	Fuzzy Logic	Digital-Image Processing	Wireless Networks
5	Advance Computer Architecture	CAD of Digital Systems	Embedded Systems	Pattern Recognition
6	Software Project Management	CAD of Digital Systems	Digital-Image Processing	Pattern Recognition

Table III: Elective enrollment database after mapping

Student Enrollment number	E11	E12	E13	E21	E22	E23	E31	E32	E33	E41	E42	E43
1	1	0	0	1	0	0	0	0	1	0	0	1
2	0	1	0	0	0	1	0	1	0	0	1	0
3	0	0	1	1	0	0	1	0	0	0	1	0
4	1	0	0	0	0	1	0	0	1	1	0	0
5	0	1	0	0	1	0	0	1	0	0	0	1
6	0	0	1	0	1	0	0	0	1	0	0	1

In general, if the interval size is *N* and the number of electives is *NE*, assuming that all items are large, the number of candidates 2-itemsets that are not generated with the new algorithm are
$= |C_2^{prune}| = NE * N (N\text{-}1)/2$
The total number of candidate 2-itemsets generated with *apriori* algorithm, assuming that all items are large = $|C_2^{apriori}| = NL_1 * (NL_1\text{-}1)/2$

Where $NL_1 = NE * N$ is the number of large items.

Thus the new algorithm generates $|C_2^{new}| = |C_2^{apriori}| - |C_2^{prune}|$ candidate 2-itemsets

If all the fields of the example dataset in Table III are large, apriori based algorithms will generate 66 candidate 2-itemsets, whereas using our algorithm the following twelve candidate 2-itemsets are pruned and the remaining 54 candidate 2-itemsets are generated.

E11 (E12+E13) + E12E13 + E21 (E22+E23) + E22E23 + E31 (E32+E33) + E32E33+ E41 (E42+E43) + E42E43

With some configuration of the student elective course enrollment dataset, if the large 2-itemsets L_2 are:

{E11E21, E11E22, E11E32, E11E33, E11E41, E12E22, E12E32, E21E31, E21E32, E21E43, E31E41, E31E42}

R_1= {E11, E12}, R_2= {E21, E22}, R_3= {E31, E32, E33}, R_4= {E41, E42, E43}

From the domain knowledge it is known that the support of candidate 3-itemsets like E11E21E22 will be zero, as the courses E21 and E22 are from same elective group. So we do not generate such itemsets in step-4 of the algorithm in Fig. 1.

The candidate 3-itemsets that are generated are: {E11E21E32, E11E21E33, E11E21E41, E11E22E32, E11E22E33, E11E22E41, E11E32E41, E11E33E41, E12E22E32, E21E31E43, E21E32E43} and the rest of the candidate 3-itemsets that otherwise will be generated if *Apriori* algorithm is used are pruned. The same approach of reducing the number of candidates can be followed for higher sized candidates. Thus reduction in the generated candidate itemsets restricts the search space improving the performance of association rule mining. The domain knowledge is incorporated for reducing the number of candidate sets generated and counted in every iteration *k* of the *Apriori* algorithm. We refer the developed algorithm as *Apriori_Domain*.

VI. EXPERIMENTAL RESULTS

To see the performance gain achieved by using domain knowledge in mining algorithms, we compare the execution time of *Apriori* and *Apriori_Domain* on Intel P4 1.6 GHZ processor with 40GB of hard disk and 256 MB of main memory. The implementation of *Apriori* is available at [20]. *Apriori_Domain* for mining association rules is developed in C by incorporating the domain knowledge to *Apriori* implementation. The experiments have been performed on synthetic data generated by us.

A. Test Data

We generate two synthetic datasets for simulating elective course enrollment databases. Each row of the generated dataset contains the list of subjects chosen by a student. The routine developed for generating these datasets takes, *n* the number of elective groups, m_i (for i = 1 to *n*) the number of subjects available for choosing from each of the *n* elective groups and *r* the number of students (tuples) enrolled in the university. The characteristics of the datasets generated are shown in Table IV. For the experimental purposes we have taken equal number of subjects offered in each elective group.

Table IV: Synthetic Dataset characteristics

Dataset	*n*	*m*	*r (in lakhs)*	Size in Mbytes
n7m5r1.5	7	5	1.5	2.9
n8m6r2	8	6	2	4.48

B. Relative Performance of Apriori and Apriori-Domain

The performance of the two algorithms *Apriori* and *Apriori_Domain* has been compared with the execution time taken by these for mining frequent itemsets for different minimum support values. The execution time in seconds taken by *Apriori* and *Apriori_Domain* for mining frequent itemsets from the *n7m5r1.5* and *n8m6r2* datasets is shown in Table V and Table VI respectively for different values of minimum support. The comparison of the algorithms is shown graphically in Fig. 1 and Fig. 2. The x-axis represents the minimum support value in percentage and the y-axis is the corresponding execution time. From the experimental results it is evident that the *Apriori_Domain* algorithm that uses the domain knowledge outperforms the *Apriori* algorithm that does not incorporates the domain knowledge.

VII. CONCLUSION

An essential research issue surrounding association rule mining is to find fast and effective association rule mining algorithms. In this paper we have shown the use of domain knowledge in data mining. We have developed an algorithm that makes use of domain knowledge for efficient data mining of association rules. Experimental results show that the developed algorithm results in faster mining of association rules from the elective course university dataset as compared to mining the same patterns with an association rule-mining algorithm that does not makes use of domain knowledge.

Table V: Execution times on *n7m5r1.5*

minsup (%)	Execution time (seconds)	
	Apriori	Apriori_Domain
.14	289.59	264.38
.12	407.75	367.24
.1	543.54	482.33
.08	664.82	579.6
.06	786.48	684.55
.04	1082.93	931.12
.02	2036.37	1665.82

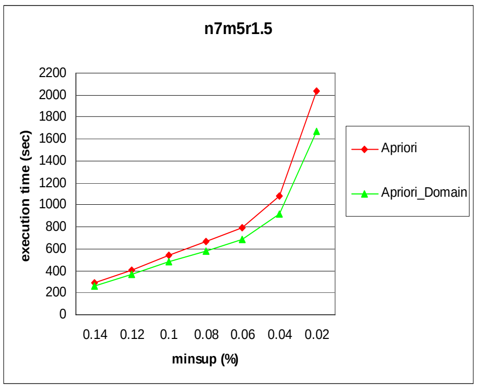

Fig. 1: Performance on *n7m5r1.5*

Table VI: Execution times on *n8m6r2*

minsup (%)	Execution time (seconds)	
	Apriori	Apriori_Domain
.5	166.47	163.8
.25	350.85	334.3
.15	687.24	649.37
.1	1081.15	1013.14
.05	1853.16	1705.75
.04	2019.06	1855.03

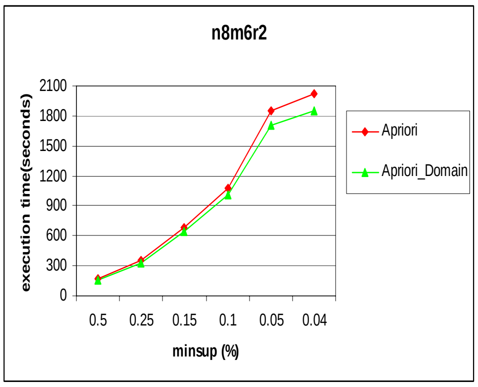

Fig. 2: Performance on *n8m6r2*

REFERENCES

1. Rakesh Agrawal, Tomaz Imielinski and Arun Swami, “Mining Association Rules between Set of Items in Large Databases,” *In Proceedings of the ACM SIGMOD Conference on Management of Data (SIGMOD 1993),* pages 207-216, May 1993.

2 Rakesh Agrawal and R. Srikanth, “Fast Algorithm for Mining Association Rules,” *In Proceedings of 20th International Conference on Very Large Databases (VLDB 1994)*, Santiago Chile, September 1994.

3 Brachman R. Anand T., “The process of Knowledge Discovery in Databases: A Human Centered Approach,” *Advances in Knowledge discovery & Data Mining, AAAI Press & the MIT Press:* California, 996, (1996), 37-57.

4 Yoon S., Henschen L. J., Park E.K., Makki S., “Using Domain Knowledge in Knowledge Discovery,” *Proceedings ACM conference CIKM 1999, Kansas city, MO, USA*, page 243-250.

5 Tseng, Shin-Mu, “Mining Association Rules with Interestingness Constraints in Large Databases,” *International Journal of Fuzzy Systems*, Vol. 3, No. 2, June 2001.

6 Ioannis Kopanas, N. Avouris, S. Daskalaki, “The role of domain knowledge in a large scale Data Mining Project,” *In Methods and Applications of Artificial Intelligence, Lecture notes in AI,* page 288-299, Springer-Verlag, Berlin, 2002.

7 Raymond C.W Wong and Ada W.C Fu, “ISM: Item Selection for Marketing with Cross-Selling Considerations,” *In Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, Australia, 2004.

8 Savasere, E. Omiecinski, and S. Navathe, “An Efficient Algorithm for Mining Association Rules in Large Databases,” *In Proceedings of the 21st International conference on Very Large Databases (VLDB 1995),* pages 432-443, Zurich, Switzerland, September 1995.

9 Hannu Toivonen, “Sampling Large Databases for Association Rules,” *In Proceedings of the 22nd International conference on Very Large Databases (VLDB 1996)*, pages 134-145, Bombay, India, September 1996.

10 Sergy Brin, Rajeev Motwani, J.Ullman and Sergey Tsur, “Dynamic Itemset Counting and Implication Rules for Market Basket Data,” *In Proceedings of ACM SIGMOD International conference on Management of Data (SIGMOD 1997),* pages 255-264, Arizona, USA, May 1997.

11 Brian Dunkel, Nandit Soparkar, “Data Organization and Access for Efficient Data Mining,” *In Proceedings of the 15th International conference on Data Engineering (ICDE* 1999), pages 522-529, Sydney, Australia, 1999.

12 Zahir Tari, Wensheng Wu, “ARM++ - A Hybrid Association Rule Mining Algorithm,” *RMIT Technical Report*, Department of Computer Science, TR-99-1, 1999.

13 Jiawei Han, Jiian Pei and Yiwen Yin, “Mining frequent patterns without Candidate generation,” *In International Conference on Management of Data (SIGMOD 2000)*, Dallas, May 2000.

14 P. Shenoy, J.Haritsa, S.Sudarshan, G.Bhalotia, M.Bawa, D.Shah, “Turbo Charging Vertical Mining of Large Databases,” *In Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD 2000),* pages 22-33, Dallas, Texas, May 2000.

15 Rakesh Agrawal, Charu Agrawal and V.V. Prasad, “Depth First Generation of Long Patterns,” *In proceedings of 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* pages 108-118, August 2000.

16 Mohammed Zaki, Karan Gouda, “Fast Vertical Mining Using Diffsets,” *In Technical report 01-1*, Department of Computer Science, Rensselaser Polytechnic Institute, March 2001.

17 S.Orlando, P.Palmerini, R.Perego, “Enhancing the Apriori Algorithm for Frequent Set Counting,” *In Proceedings of 3rd International Conference on Data Warehousing and Knowledge Discovery* (*DaWaK 2001)*, Munich, September 2001.

18 S. Orlando, P. Palmerini, R. Perego and F. Silvestri, “Adaptive and Resource Aware Mining of Frequent Sets,” *In Proceedings of the IEEE International Conference on Data Mining (ICDM 2002),* pages 338 - 345, December 2002.

19 B. Goethals, “Survey on Frequent Pattern Mining,” *Technical report*, Helsinki Institute of Information Technology, 2003.

20 Website of Dr. Tingshao Zhu
http://www.cs.ualberta.ca/~tszhu