Computer Science in Education

Irshad Ullah

Institute, Computer Science, GHSS Ouch
Khyber Pakhtunkhwa, Chinarkot, ISO 2-alpha – PK, Pakistan

Abstract

Computer science or computing science (sometimes abbreviated CS) is the learning of the theoretical foundations of information and computation, and of practical techniques for their execution and application in computer systems. It is often described as the efficient study of algorithmic processes that produce, explain, and transform information.

In this work, I use Data Mining algorithms from the field of computer science for the analysis process to prove experimentally and practically that how reliable, efficient and fast are these for the analysis of data in education? A solid mathematical threshold (0 to 1) is set to analyze the data. The obtained results will be tested by applying the approach to the databases and data warehouses of different sizes with different threshold values. The results produce will be of different strength from short to the largest sets of data list. By this, we may get the results for different purposes e.g. making future education plan.

Key Words of the abstract
Computer Science, Education, Results, Education Plan.

1 Introduction

1.1 Computer Science

Computer science or computing science (sometimes abbreviated CS) is the study of the theoretical foundations of information and computation, and of practical techniques for their implementation and application in computer systems.[1][2]

1.2 Data Mining

Data Mining is the discovery of hidden information found in databases [14] [19]. Data mining functions include clustering, classification, prediction, and associations. One of the most significant data mining applications is that of mining association rules. Association rules, first introduced in 1993 [17], are used to identify relationships among a set of items in databases. The AIS algorithm is the first published algorithm developed to generate all large itemsets in a transaction database [17]. This algorithm has targeted to discover qualitative rules. This technique is limited to only one item in the consequent. This algorithm makes multiple passes over the entire database. The SETM algorithm is proposed in [13] and motivated by the desire to use SQL to calculate large itemsets [18]. In this algorithm each member of the set large itemsets, Lk, is in the form <TID, itemset> where TID is the unique identifier of a transaction. Similarly, each member of the set of candidate itemsets, Ck, is in the form <TID, itemset>. Similar to [17], the SETM algorithm makes multiple passes over the database.

The Apriori algorithm [16] is a great achievement in the history of mining association rules. It is by far the most well-known association rule algorithm. This technique uses the property that any subset of a large itemset must be a large itemset.

The Off-line Candidate Determination (OCD) technique is proposed in [10], and is based on the idea that small samples are usually quite good for finding large itemsets.

Sampling [9] reduces the number of database scans to one in the best case and two in the worst. A sample which can fit in the main memory is first drawn from the database. The set of large itemsets in the sample is then found from this sample by using a level-wise algorithm such as Apriori.

Each association rule mining algorithm assumes that the transactions are stored in some basic structure, usually a flat file or a TID list, whereas actual data stored in transaction databases is not in this form. All approaches are based on first finding the large itemsets. The Apriori algorithm appears to be the center of all the association rule mining algorithms. In this work my focus is on association rule mining technique. I take two algorithms, first the well known Apriori and then our own developed SI [12] algorithm.

2. Association Rule Problem

A formal statement of the association rule problem is as follows:

Definition: [17] [6] Let I = {i_1, $i_{2,}$…, i_m} be a set of m distinct attributes. Let D be a database, where each record (tuple) T has a unique identifier, and contains a set of items such that T$\subseteq$ I. An association rule is an implication of the form of X $\Rightarrow$Y, where X, Y $\subseteq$I are sets of items called itemsets, and X $\cap$Y =ϕ . Here, X is called antecedent while Y is called consequent; the rule means X $\Rightarrow$Y. Association rules can be classified based on the type of vales, dimensions of data, and levels of abstractions involved in the rule. If a rule concerns associations between the presence or absence of items, it is called Boolean association rule. And the dataset consisting of attributes which can assume only binary (0-absent, 1-present) values is called Boolean database.

3. Logical Data Analysis

The logical analysis of data was originally developed for the analysis of datasets whose attributes take only binary (0-1) values [4, 5, 8].Since it turned out later that most of the real-life applications include attributes taking real values, a “binarization" method was proposed in [3]. The purpose of binarization is the transformation of a database of any type into a “Boolean database”.

Table 1.Original Database

ID	Age 21….28	Age 30…40	M.Status Single	M.Status Married
1	1	0	1	0
2	0	1	0	1

LAD is a methodology developed since the late eighties, aimed at discovering hidden structural information in Boolean databases. LAD was originally developed for analyzing binary data by using the theory of partially defined Boolean functions. An extension of LAD for the analysis of numerical data sets is achieved through the process of “binarization” consisting in the replacement of each numerical variable by binary “indicator” variables, each showing whether the value of the original variable is present or absent, or is above or below a certain level. LAD has been applied to numerous disciplines, e.g. economics and business, seismology, oil exploration, medicine etc. [15].

3.1 Binarization

The methodology of LAD is extended to the case of numerical data by a process called binarization, consisting in the transformation of numerical (real valued) data to binary (0, 1) ones. In this [7] transformation we map each observation u = (uA, uB,…) of the given numerical data set to a binary vector x(u) = (x1, x2,…) Є {0, 1}n by defining e.g. x1 = 1 iif uA ≥ α1, x2 = 1 iif uB ≥ α2, etc, and in such a way that if u and v represent, respectively, a positive and negative observation point, then x(u) ≠ x(v). The binary variables xi, i = 1,2, …, n associated to the real attributes are called indicator variables, and the real parameters αi, i = 1, 2, …, n used in the above process are called cut points.

The basic idea of binarization is very simple. It consists in the introduction of several binary attributes associated to each of the numerical attributes; each of these binary attributes is supposed to take the value 1 (respectively, 0) if the numerical attribute to which it is associated takes values above (respectively, below) a certain threshold. Obviously the computational problem associated to binarization is to find a minimum number of such threshold values (cut points) which preserve the essential information contained in the dataset.

In order to illustrate the binarization of business datasets, let us consider the examples presented in Table 1. A very simple binarization procedure is used for each variable “age” and “marital status”. Quantitative attributes such as “age” is divided into different ranges like age: 20..29, 30..39, etc. The “marital status” variable is divided into binary values by converting its domain values into attributes.

Table 2 Boolean Database

ID	Age	M.Status	#cars
1	22	Single	0
2	37	Married	2

3.2 Binary Variables

A binary variable has only two states: 0 or 1, where 0 means that the variable is absent, and 1 means that it is present. If all binary variables are thought of as having the same weight, we have the 2-by-2 contingency table of table 3, where a is the number of variables that equal 1 for both items i and j, b is the number of variables that equal 1 for item i but that are 0 for item j, c is the number of variables that equal 0 for item i but equal 1 for item j, and d is the number of variables that equal 0 for both item i and j. The total number of variables is z, where z = a + b + c + d.

Table 3. A contingency table for binary variables

	Item j			
Item i		**1**	**0**	**Sum**
	1	A	B	a + b
	0	C	D	c+ d
	Sum	A + c	b+ d	Z

For noninvariant similarities, the most well-known coefficient is the Jaccard dissimilarity coefficient, where the number of negative matches d is considered unimportant and thus is ignored in the computation:

$$d(I,J) = \frac{b+c}{a+b+c} \quad \text{Eq. (3.2.1)}$$

The measurement value 1 suggests that the objects i and j are dissimilar and the measurement value 0 suggests that the objects are similar. This method is used in SI algorithm while the Apriori algorithm works using similarity measures.

3.3 SI Algorithm

I use two algorithms for the implementation purpose. Our develop SI algorithm and the well known Apriori algorithm to check the accuracy and efficiency.

Input
Φ User specified threshold between 0 And 1
T Binary transactional Database
Output
Frequent itemsets
Step

p = { i_1,i_2,…..in} set of data items in transactional database.

Create K Map for all the permutation in row.

Scan the transactional database and put the presence for every combination of data items in corresponding K Map for every permutation of row.

For every permutation of p:

a) Calculate dissimilarity using K Map Constructed for every permutation using the following Jacquard's dissimilarity equation.

d (i1,i2,…..in) = $\sum_{i1=0}^{1} \sum_{i2=0}^{1} \sum_{i3=0}^{1}$,………. $\sum_{in=0}^{1}$ f(i_1,i_2,…..in)f(0,0,…..0) – f(1,1,……1)/ f $\sum_{i1=0}^{1} \sum_{i2=0}^{1} \sum_{i3=0}^{1}$,…… $\sum_{in=0}^{1}$ f(i_1,i_2,…..in) - f(0,0,…..0). Eq. (3.3.1)

b) If d < Φ then i_1,i_2,…..in are frequent .
Eq. (3.3.2)

3.4 Apriori Algorithm

Input
Φ User specified threshold between 0 And 100
T Binary transactional Database
Output
Frequent lists

Apriori algorithm work on similarity measure while the SI algorithm works on dissimilarity measure.

4. Experimental Results

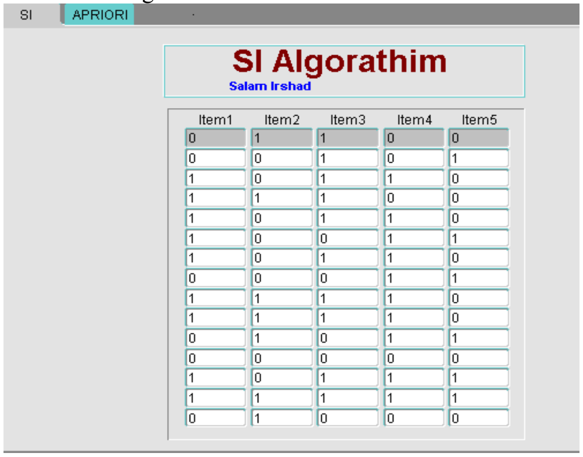
I performed different experiments to check the results and efficiency of the technique. The data required in database should be in binary format. I downloaded the dataset transa from the net [11]. The data was stored in a format:

0 1 0 0 1 0 0 1

I coded the algorithms in ORACLE 10g using laptop computer having 20GB hard drive and 1.6MH processor. I create a table in the database to store the data for the purpose of experiment. To load the data to the database oracle provides a facility by making a control file and then by using SQL loader. We first convert the data into a format that the item now is separated by commas instead of spaces. Now the data is loaded to the table with the help of SQL loader and look like

0,1,0,0,1,0,0,1

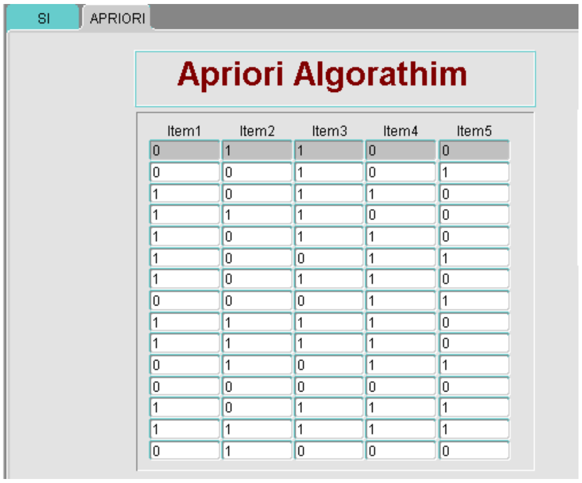
After loading the data into table the algorithms are implemented on the database having fifty records, initially.



SI Algorathim
Salam Irshad

Item1	Item2	Item3	Item4	Item5
0	1	1	0	0
0	0	1	0	1
1	0	1	1	0
1	1	1	0	0
1	0	1	1	0
1	0	0	1	1
1	0	1	1	0
0	0	0	1	1
1	1	1	1	0
1	1	1	1	0
0	1	0	1	1
0	0	0	0	0
1	0	1	1	1
1	1	1	1	1
0	1	0	0	0

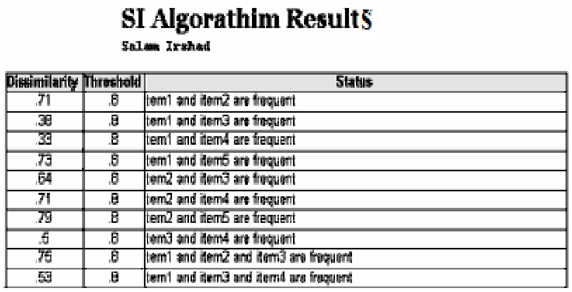
Figure 1.SI Algorithm



Apriori Algorathim

Item1	Item2	Item3	Item4	Item5
0	1	1	0	0
0	0	1	0	1
1	0	1	1	0
1	1	1	0	0
1	0	1	1	0
1	0	0	1	1
1	0	1	1	0
0	0	0	1	1
1	1	1	1	0
1	1	1	1	0
0	1	0	1	1
0	0	0	0	0
1	0	1	1	1
1	1	1	1	1
0	1	0	0	0

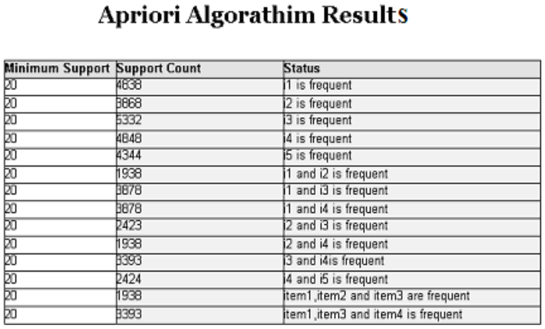
Figure 2. Apriori Algorithm



SI Algorathim Results
Salam Irshad

Dissimilarity	Threshold	Status
.71	.8	item1 and item2 are frequent
.38	.8	item1 and item3 are frequent
.33	.8	item1 and item4 are frequent
.73	.8	item1 and item5 are frequent
.64	.8	item2 and item3 are frequent
.71	.8	item2 and item4 are frequent
.79	.8	item2 and item5 are frequent
.5	.8	item3 and item4 are frequent
.75	.8	item1 and item2 and item3 are frequent
.53	.8	item1 and item3 and item4 are frequent

Figure 3. SI Algorithm results



Apriori Algorathim Results

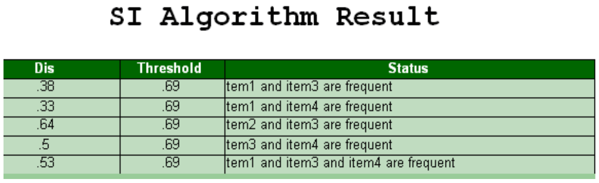
Minimum Support	Support Count	Status
20	4838	i1 is frequent
20	3868	i2 is frequent
20	5332	i3 is frequent
20	4848	i4 is frequent
20	4344	i5 is frequent
20	1938	i1 and i2 is frequent
20	3878	i1 and i3 is frequent
20	3878	i1 and i4 is frequent
20	2423	i2 and i3 is frequent
20	1938	i2 and i4 is frequent
20	3393	i3 and i4is frequent
20	2424	i4 and i5 is frequent
20	1938	item1 ,item2 and item3 are frequent
20	3393	item1 ,item3 and item4 is frequent

Figure 4.Apriori Algorithm results

The largest frequent lists generated by the algorithm are
I_1, $I_{2,}$ I_3 ,, I_1, I_3, I_4
After giving the data to Apriori algorithm it also produce the same results. With the same largest frequent sets contain,
I_1, $I_{2,}$ I_3
I_1, I_3, I_4
After loading more data, the total records in the database became 500. Applying Apriori and SI algorithms on the updated database, the results produced are given.
The largest frequent list.
I_1, $I_{2,}$ I_3
I_1, I_3, I_4
We found that again the algorithms produce the same results. After loading more data the total number of records become 2000. And again applying the algorithms on the database, the results produce are given below.
I_1, $I_{2,}$ I_3
I_1, I_3, I_4
So this is again that the algorithms produce the same results. After loading more data to the database the total records in the database are 4000. Again applying Apriori and SI algorithm on the database the results produced are given.
I_1, $I_{2,}$ I_3
I_1, I_3, I_4

We see that again the algorithms produce the same results. Now we have to load more data to the database the total number of records become 8000. And once again applying both the algorithm on the database the results produce are given below.
I_1, $I_{2,}$ I_3
I_1, I_3, I_4
We see that again the algorithm produces the same results.
Up to this we analyze the performance, efficiency and accuracy of the algorithms by changing size of the database. And this is clear

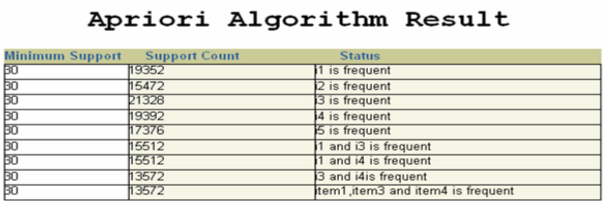
that the results produce from this database are very consistent and reliable for the evaluation of learners, tutors and subjects etc. After this we change the input threshold to analyze the performance, efficiency and accuracy at different threshold values. The input threshold changes from .80% to .70% (dissimilarity) for SI algorithm and from 20% to 30% (similarity) for Apriori algorithm. The database contains 8000 records and after applying both the algorithms the results produces are given below.



SI Algorithm Result

Dis	Threshold	Status
.38	.69	tem1 and item3 are frequent
.33	.69	tem1 and item4 are frequent
.64	.69	tem2 and item3 are frequent
.5	.69	tem3 and item4 are frequent
.53	.69	tem1 and item3 and item4 are frequent

Figure 5 SI Algorithm Results



Apriori Algorithm Result

Minimum Support	Support Count	Status
30	19352	i1 is frequent
30	15472	i2 is frequent
30	21328	i3 is frequent
30	19392	i4 is frequent
30	17376	i5 is frequent
30	15512	i1 and i3 is frequent
30	15512	i1 and i4 is frequent
30	13572	i3 and i4is frequent
30	13572	item1,item3 and item4 is frequent

Figure 6. Apriori Algorithm Results

The largest frequent list produce is I_1, I_3, I_4
Now this is clear that both algorithms produced the same results at different threshold. Up to this, we analyze that these techniques are very reliable for the analysis and discovery of hidden pattern and information in any type of database just like in this educational database.

5 Graphical analysis

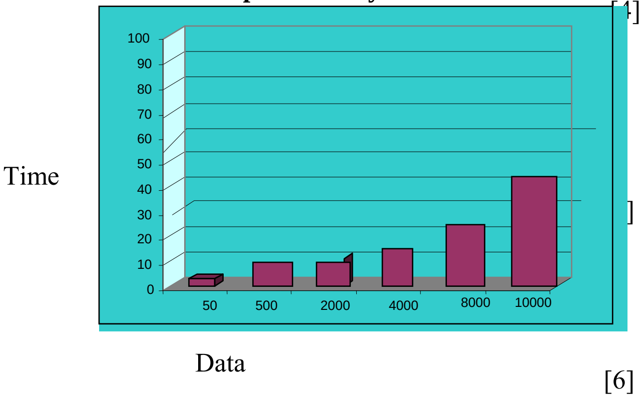


Figure 7. Graphical analysis of the results

6 Concluding Remarks And Future Work.

In this research, I study that how data mining techniques are used for the evaluation and Knowledge discovery in the field of education. The output produced was based on realistic reasons and values so it is reliable, efficient and precise for the experts. Here we produce the results by performing different experiments and prove that such techniques are very consistent. On the basis of the output we may predict and evaluate students and teachers. And also we take the results to make categories of the learners. By this we may enhance the learning process. The subjects may be divide into groups on the basis of similarity, dissimilarity measure. Further we may perform experiments for other algorithms from different point of view on different data storage.

7 References

[1] Comer, D. E.; Gries, D., Mulder, M. C., Tucker, A., Turner, A. J., and Young, P. R. "Computing as a discipline". Communications of the ACM . (Jan. 1989) vol32 (1)

[2] Wegner, P. "Research paradigms in computer science". Proceedings of the 2nd international Conference on Software Engineering. San Francisco, California, United States: Press, Los Alamitos, CA. (October 13–15, 1976).

[3] Boros E., P.L. Hammer, T. Ibaraki, A. Kogan. Logical Analysis of Numerical Data. Mathematical Programming, (1997). 79:163-190.

4] Boros E., P.L. Hammer, T. Ibaraki, A. Kogan, E. Mayoraz, I. Muchnik. An Implementation of Logical Analysis of Data. IEEE Transactions on knowledge and Data Engineering, 12(2) (2000):292-306.

] Crama Y., P.L. Hammer, T. Ibaraki. Cause-effect Relationships and Partially Defined Boolean Functions. Annals of Operations Research, 16(1988).:299-325.

[6] David Wai-Lok Cheung, Vincent T. Ng, Ada Wai-Chee Fu, and Yongjian Fu.. Efficient Mining of Association Rules in Distributed Databases, IEEE Transactions on Knowledge and Data

Engineering, (December 1996)Vol. 8, No. 6, pp. 911-922.

[7] E. Boros, P. L. Hammer, T. Ibaraki, A. Kogan, E. Mayoraz and I. Muchnik An implementation of logical analysis of data, RUTCOR Research Report RRR 22-96, Rutgers University, 1996.,pp. 911-922.

[8] Hammer P.L. The Logic of Cause-effect Relationships, Lecture at the International Conference on Multi-Attribute Decision Making via Operations Research-based Expert Systems, (1986)Passau, Germany.

[9] Hannu Toivonen. Sampling Large Databases for Association Rules, Proceedings of the 22nd International Conference on Very Large Databases, (1996) pp. 134-145, Mumbai, India.

[10] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo Efficient Algorithms for Discovering Association Rules, Proceedings of the AAAI Workshop on Knowledge Discovery in Databases (KDD-94), (July 1994).pp. 181-192.

[11]
http://www2.cs.uregina.ca/~dbd/cs83
1/
notes/itemsets/item set prog1.htm

[12] Irshad Ullah, Abdus Salam and Saif-ur-Rehman ,Dissimilarity Based Mining for Finding Frequent itemsets. Proceedings of 4th international conference on statistical sciences (2008) Volume (15), University of Gujrat Pakistan, 15: 78

[13] M. Houtsmal and A.. Set-Oriented Mining for Association Rules in Relational Databases, Proceedings of the 11th IEEE International Conference on Data Engineering, Swami (1995)pp. 25-34, Taipei,Taiwan.

[14] Ming-Syan Chen, Jiawei Han and Philip S. Yu.. Data Mining: An Overview from a Database Perspective, IEEE transactions on Knowledge and Data Engineering, (1996) Vol. 8, No. 6, pp. 866-883

[15] Peter L. Hammer Tiberius Bonates.. Logical Analysis of Data: From Combinatorial Optimization to Medical Applications, RUTCOR Research Report(2005) RRR 10 - 2005.

[16] Rakesh Agrawal and Ramakrishnan. Fast Algorithms for Mining Association Rules in Large Databases, Proceedings of the Twentieth International Conference on Very Large Databases, Srikant.(1994)pp. 487-499, Santiago, Chile.

[17] R. Agrawal, T. Imielinski, A. Swami. Mining Associations between Sets of Items in Massive Databases, Proc. of the ACM-SIGMOD Int'l Conference on Management of Data,Washington D.C. (1993).

[18] Ramakrishnan Srikant. Fast Algorithms for Mining Association Rules and Sequential Patterns,Ph. D. Dissertation,University of Wisconsin, Madison. (1996)

[19] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From Data Mining to knowledge Discovery: An Overview, Advances in Knowledge Discovery and Data Mining, (1996). AAAI Press, , pp 1-34.

Author: **Irshad Ullah**

MS-CS (Database systems)

2006, 2007.

B.Ed 2004

M.SC Computer Science 2002

B.SC Computer Science 2000

Senior IT Teacher (**SITT**) Since

2004.

Research Interest: Data Mining

Member of the: **ISOSS.**

Two publications and presentations

an **International Conferences**.