REDUCTION OF NUMBER OF ASSOCIATION RULES WITH INTER ITEMSET DISTANCE IN TRANSACTION DATABASES

Pankaj Kumar Deva Sarma[1] and Anjana Kakati Mahanta[2]

[1]Department of Computer Science, Assam University, Silchar, Assam, India
PIN-788011
pankajkdsarma@yahoo.com, pankajgr@rediffmail.com
[2]Department of Computer Science, Gauhati University, Guwahati, Assam, India
PIN - 781014
anjanagu@yahoo.com

ABSTRACT

Association Rule discovery has been an important problem of investigation in knowledge discovery and data mining. An association rule describes associations among the sets of items which occur together in transactions of databases.The Association Rule mining task consists of finding the frequent itemsets and the rules in the form of conditional implications with respect to some prespecified threshold values of support and confidence.The interestingness of Association Rules are determined by these two measures. However, other measures of interestingness like lift and conviction are also used. But, there occurs an explosive growth of discovered association rules and many of such rules are insignificant. In this paper we introduce a new measure of interestingness called Inter Itemset Distance or Spread and implemented this notion based on the approaches of the apriori algorithm with a view to reduce the number of discovered Association Rules in a meaningful manner. An analysis of the working of the new algorithm is done and the results are presented and compared with the results of conventional apriori algorithm.

KEYWORDS

Association rules, frequent itemsets, support, confidence, Inter itemset distance, spread, data mining.

1. INTRODUCTION

Data Mining or Knowledge Discovery in Databases came to being as a field of research since the very early 1990s and since it has grown tremendously with wide ranging applications from Business, Science and Engineering Research, Medical Diagnosis, DNA and Genome data analysis. Data Mining concerns with design and development of computational algorithms and techniques for discovering hidden patterns and rules which are nontrivial, interesting, previously unknown and potentially useful from data in databases [1]. These discovered rules are immensely useful in decision making.

The problem of Association Rule Mining is to discover a set of items shared among a large number of transactions in a database. In association rule discovery, it is found how the presence of a set of items in a transaction influences the presence of another set of items in the same transaction. For example, let us consider the database of daily issue and return of books of a

DOI: 10.5121/ijdms.2012.4505

particular library, where the transactions represent the subscribers of the Library and the attributes or items represent the books. Here while the subscribers get books issued in their names and returns, these are recorded as transactions in the database of issue and return. The volume of transactions thus generated can become large over a period of time and stores the subscribers' usage pattern of the library. However, these patterns cannot be discovered by conventional means and needs relevant data mining technique to discover the same. One of the discovered patterns could be the set of books which are most frequently issued or returned together by the subscribers. For example, 50% of the subscribers who issue Ulman's *Theory of Automata* also issue *Compilers* by the same author. The library can utilize this knowledge and many such for better service to the subscribers. The problem is to discover all such association rules in large databases. Efficient algorithms are required to be designed with appropriate measure of interestingness.

Further, there are issues related to measure of interestingness of association rules. Support and confidence are the two most widely used mearures. Other parameters include correlation or lift or interest and conviction. The concept of correlation rule is introduced in [5]. A correlation rule is defined as a set of itemsets that are correlated. The motivation for developing such a rule is that negative correlations may be useful [6] [56]**.** Correlation satisfies upward closure in the itemset lattice. Thus if a set is correlated then every superset of it is also correlated. For an association rule of the form A => B in a transaction database, where A and B are the itemsets, *support* of an itemset is defined as the frequency of its occurrence in the database. The *support* (s) (expressed as percentage) of the rule A => B is the probability of occurrence of the itemset (A U B) and is given by

$$s\ (A \Rightarrow B) = P(A\ U\ B) \qquad (1)$$

The *confidence* (c) of the rule A => B is defined as

$$c\ (A \Rightarrow B) = \text{support}(A\ U\ B)\ /\ \text{support}\ (A) = P(A\ U\ B)\ /\ P(A) \qquad (2)$$

The value of support and confidence measures the strength of a rule. Chi squared test for independence is another measure for significance of rules [6] [56]**.** The Chi squared significance test takes into account both presence or absence of items in sets. On the other hand support and confidence are the measures which are based only on the presence of items in the sets. The task of discovering interesting Association Rules from large databases computationally intensive. The search space is exponential in terms of the number of distinct database items present in the transaction database. Further, there are billions of database transactions for which there occurs shortage of main memory to accommodate the transactions of the large databases. This increases the data transfer activity. Most of the approaches of Association Rule Mining require multiple database scans, which is expensive[2]. Alternatively, Parallel Algorithms [3] and algorithms based on sampling [4] are also designed and implemented. The method of sampling was used to contain the explosive growth of the search space. However, the methods based on sampling do not always get the true representation of the data to work upon can be sensitive to the data - skew which may adversely affect the performance [2]. The parallel algorithms have the additional overhead of data transfer and message passing which may consume longer time than desired. Research issues related to the association rule mining algorithms include scalability, exponential growth of the search space and the increase in discovered rules with the increase in the number of items in the database, multiple database scans and I/O reduction, reduction of the database scans and so on.

The rest of the paper is organized as follows. In section 2, we introduce the concept of Average Inter Itemset Distance or Spread as new measure of interesingness for reducing the rules. The association rule discovery problem is described in section 3. Related works are analysed in section 4. Mining association rules with average inter itemset distance or spread is discussed in section 5. In section 6, the algorithmic steps and its analysis for the mining association rules based on average inter itemset distance along with support and confidence is given. In section 7, the working of the algorithm is demonstrated with an example. In section 8, implementation and results of the proposed algorithm is presented. In section 9, conclusion and future scope of works is discussed.

2. INTER ITEMSET DISTANCE (SPREAD) AS A MEASURE OF INTERESTINGNESS

Support and confidence are by far the most widely used and popular measures of interestingness. Confidence is similar to the conditional probability of occurrence of a rule. To find the support of an itemset, the database is scanned and only the frequency of occurrence of the itemset is calculated irrespective of the position of occurrence of the itemset in the transactions. The position of occurrence of an itemset is according to the Transaction IDs (TID) of the transactions in which the itemset is found to be present and there are number of transactions in which the itemset may not have been occurred. These transactions of nonoccurrence of the itemset are of importance for the relative patterns of occurrences of the itemsets while mining association rules. Such transactions in which an itemset has not occurred are calculated and the discovered association rules are reinforced with additional meaning with the existing measures support and confidence. It is observed that the measures support, confidence, correlation etc. for association rules as mentioned above do not take into account the number of intervenning transactions between two consecutive occurrences of an itemset from which the rules are generated in the database of transactions. When the support of an itemset is counted to determine whether the itemset is frequent or not, only the overall occurrences of each itemset is counted in the whole transaction database. From the values of support count of the itemsets, it is not possible to know how these itemsets are distributed with respct to one another in the whole transaction database. Therefore, it is required to find out after one occurrence of an itemset in the transaction database, how many transactions are there in the middle in which the same itemset has not occurred before its next occurrence and so on. In this manner, if continued then there are several intervenning transactions for the occurrences of an itemset till its last occurrence. All these intervenning number of transactions separating the consecutive occurrences of an itemset, when added together and divided by the number of the gaps of nonoccurrences, then we get the average separation in terms of the number of transactions in which the itemset has not occurred between the first and the last occurrences of the itemset. This, we call **Average Inter Itemset Distance (IID) or Spread** for an itemset.

The value of *Average Inter Itemset Distance or Spread* gives an indication how closely or sparsely the itemsets in the transaction database are separated from each other within its lifespan. The *lifespan* of an itemset is the number of transactions including the first and the last transactions of occurrence of the itemset which need not necessarily always be the first and the last transactions of the database. Together with the support, average inter itemset distance or spread is used as another measure for an association rule generated from a frequent itemset. By using threshold values for average inter itemset distance in addition to the threshold values for support and confidence, association rules can be discovered. The smaller the value of the threshold for average inter itemset distance the closure will be the spacing between the successive occurrences of an itemset. Thus with the help of a threshold for average inter itemset distance the number of frequent itemsets and hence the number of association rules genetrated can be reduced. The value of the avarage inter itemset distance implies that on an average, after how many transactions the same itemset repeats itself in the transactions of the database. Therefore the occurrence patterns of an itemset which can be obtained by using the avarage inter itemset distance or spread cannot be obtained with support. The following is an example in this context.

Let us suppose in a database of 50 transactions containing items {A, B, C, D, E} the itemset {A, B, C} occurs in the transactions 1, 5, 6, 25, 33 and 42. Thus this itemset has support count 6. Another itemset {B, C, E} occurs in the transactions 5, 7, 12, 15, 17 and 19. It also has support count 6. In this case the occurrences of the itemset {B, C, E} is closure to each other as compared to the itemset {A, B, C}. Though the support of both the itemsets is the same, there is a difference in their pattern of occurrences in the whole database. This cannot be identified based on the support of an itemset. But with the introduction of average inter itemset distance or spread as another measure for a frequent itemset and the corresponding association rules, this kind of pattern of occurrences can be discovered. The parameter average inter itemset distance can be used along with support and confidence for a rule.

The itemsets which satisfy the input thresold value for average inter itemset distance or spread is called **closely spaced itemsets**. An algorithm for discovering association rules with average inter itemset distance or spread, support and confidence based on prespecified threshold values is proposed based on the apriori algorithm. The apriori algorithm has been discussed in [3] [7] [8] [9]. The apriori algorithm and the proposed algorithm are implemented and results obtained from standard datasets are presented. It is observed that there is a reduction in the number of association rules discovered based on this approach as compared to the apriori algorithm.

3. PROBLEM STATEMENT: MINING OF ASSOCIATION RULES

The problem of mining Association Rules was first introduced in [7]. It is described as follows: Let I = {i_1, i_2, i_3, … …. …. i_m} be a set of literals called items and D be a database of transactions, where each transaction T is a set of items such that T $\subseteq$I. Given an itemset X $\subseteq$I, a transaction T contains X if and only if X$\subseteq$T. In other words, I = {i_1, i_2, i_3, … …. …. i_m} is a set of attributes over the binary domain {0,1}. Each transaction T is a tuple of the database D and is represented by identifying the attributes with value 1. A unique identifier called TID is associated with each transaction. A set of items X $\subseteq$I is called an itemset.

An association rule is an implication of the form X => Y such that X $\subseteq$I, Y$\subseteq$I and X∩Y = Φ. The rule X=>Y holds in the transaction database D with confidence c if c% of the transactions in D that contain X also contain Y. The rule X=>Y has support s in the transaction database D if s% of transactions in D contains X U Y. Confidence denotes the strength of implication and support indicates the frequencies of the occurring patterns in the rule. Rules with high confidence and strong support are referred to as strong rules in large databases [7].

Given a set of transactions D, the problem of mining association rules is to generate all association rules that have certain user specified minimum support (called *minsup*) and confidence (called *minconf*). Such an association rule is also called strong rule to distinguish it from weak ones i. e. those rules which do not meet the thresholds [11]. A set of items is called an *itemset*. An itemset with k-items is called a *k-itemset*. The support count of an itemset X denoted by σ(X) is the number of transactions in which it occurs as a subset. Support of an itemset is also expressed as percentage. That is for an itemset X, its percentage support is defined as the percentage of transactions in the database D which contains X. Support count denotes the frequency of occurrence of an itemset in the database D. Given a minimum support threshold minsup, an itemset X is said to be *large* or *frequent* if its support is not less than minsup. An itemset is called **maximal** if it is not a subset of any other itemset. Then a frequent itemset is said to be a **maximal frequent itemset** if it is a frequent itemset and no superset of this is a frequent itemset. The confidence of a rule is given by σ(X U Y)/σ(X). In other words confidence of the association rule X=>Y is the conditional probability that a transaction contains Y, given that it contains X. It has been shown that the task of mining association rules can be reduced to two subproblems [7] [8] viz.

Step 1. Find all the large or frequent itemsets with respect to a prespecified minimum support. This step is computationally and I/O intensive. Given m itemsets, there can be potentially 2^m subsets, where m is an Integer and m > 0. Frequent itemset discovery has been the major focus of research.

Step 2. Generate the association rules which satisfy the threshold from the frequent itemsets discovered. This step is relatively straight forward. Rules are of the form X\Y=>Y are generated for all the frequent itemsets X, where Y $\subseteq$ X subject to the condition that the rules have at least minimum confidence.

The performance of the mining association rules is determined by the efficiency of the method to resolve the first problem in step 1 [9]. In this paper this problem is augumented with the itroduction of the average inter itemset distance or spread of an itemset as another measure of interestingness. and a method is proposed to do it without the need of further scanning of the database. This step is integrated with the step of calculating the support of the itemsets. In the example below with the itemset I = {A, B, C, D, E} the finding of frequent itemsets and association rules with support, confidence and average inter itemset distance or spread of an itemset is described.

Example 1: Let there are ten transactions in the database given in figure 1.

TID	Items
001	A B D E
002	B C E
003	A B C D E
004	A B E
005	A B C E
006	C D
007	A B
008	C D
009	B C D
010	A D

Figure 1: Database of transactions

Let us assume minsup(s) = 30% i.e. minsup(σ) = 3 and minconf (c) = 60% as the threshold for the support and confidence of the association rules to be discovered from this database. Figure 2 below shows the frequent itemsets that are present in the database along with their respective total inter itemset distances in terms of the number of intervenning transcations in which the itemset has not been present in its lifespan and the average inter itemset distance for the frequent itemsets calculated by dividing the inter itemset distance by the number of gaps which is given for an itemset by the value (support count of the itemset – 1) i.e (σ -1).

Gaps are non zero positive integers and are counted within the lifespan of an itemset. The value of gap is 1 only when an itemset occurs twice in the database of transactions irrespective of its relative position of occurrence in the transactions. The length of a gap is the number of transactions in which the itemset has not occurred between any two of its successive occurrences. Thus inter itemset distance and gap of occurrence of an itemset different. The inter itemset distance for an itemset which occurs in consecutive transactions is zero and it is non zero if there are transactions of nonoccurrence of the itemset between its successive occurrences. Inter itemset distance is nothing but the length of a gap expressed in terms of the number of consecutive transactions of nonoccurrence of the itemset between any two successive ocurrences of an itemset. Since an itemset occurs transactions of a database randomly, therefore, the lengths of various gaps between occurrences of an itemset in its lifespan in the database of transactions are not identical, hence the sum of the lengths of all these gaps or inter itemset disatnces are calculated for each frequent itemset to find the value of total inter itemset distance of an itemset in its life span and by dividing this number by the total number of the gaps the ***average inter itemset distance or spead*** of an itemset is obtained within the lifespan of the itemset. The value of the average inter itemset distance of an itemset can be either zero or a positive real number. It is zero when all the occurrences of an itemset take place in consecutive transcations of the database and the maximum value of the average inter itemset distance of an itemset can be expressed in terms of size of the transaction database and the input support count threshold σ. The itemsets which are found on the basis of input threshold for both ***support and average inter itemset distance or spread*** are called ***Closely Spaced Frequent Itemsets (CSFI) or Closely Spaced Large Itemsets (CSLI)***. *The smaller the value of average inter itemset distance or spread the more nearer will be the occurrences of an itemset in various transactions within the lifespan of the itemset in the transaction database.* In figure 3, we show some of the assocaition rules thus discovered and the respective values of support, confidence and ***average inter itemset distance or spread.***

Frequent Itemsets	**Support Count (σ) and Support (s) (%)**	**Average Inter Itemset Distance or Spread**
A	6 (60%)	4/5 (=0.8)
B	7 (70%)	2/6 (=0.33)
C	6 (60%)	2/5 (=0.4)
D	6 (60%)	4/5 (=0.8)
E	5 (50%)	0/4 (=0)
AB	5 (50%)	2/4 (=0.5)
AD	3 (30%)	6/2 (=3.0)
AE	4 (40%)	1/3 (=0.33)
BC	4 (40%)	4/3 (=1.33)
BD	3 (30%)	6/2 (=3.0)
BE	5 (50%)	0/4 (=0.0)
CD	4 (40%)	3/3 (=1.0)
CE	3 (30%)	1/2 (=0.5)
ABE	4 (40%)	1/3 (=0.33)

Figure2(a): Frequent itemsets (minsup = 30%) with respective values of average inter itemset distance or spread.

Support Count (σ) and Support (s) (%)	**Frequent Itemsets (Average Inter Itemset Distance or Spread)**
7 (70%)	B (2/6 =0.33)
6 (60%)	A (4/5 =0.8), C (2/5 =0.4), D (4/5 =0.8)
5 (50%)	E (0/4 =0), AB (2/4 =0.5), BE (0/4 =0.0)
4 (40%)	AE (1/3=0.33), BC (4/3 =1.33), CD (3/3=1.0), ABE (1/3=0.33)
3 (30%)	AD (6/2=3.0), BD (6/2=3.0), CE (1/2=0.5)

Figure 2 (b) : Frequent itemsets (minsup = 30%) with respective values of average inter itemset distance or spread.

Frequent Itemsets (Average Inter Itemset Distance or Spread)	**Association Rules (Confidence)**

AB (2/4 =0.5)
BE (0/4 =0.0)
	A => B (5/6 = 80.33%), B => A (5/7 = 70.14%)
	B => E (5/7 = 70.14%), E => B (5/5 = 100%)

AE (1/3=0.33)
BC(4/3 =1.33)
CD (3/3=1.0)
ABE(1/3=0.33)
	A => E (4/6 = 66.67%), E => A (4/5 = 80.0%)
	C => B (4/6 = 66.67%)
	C => D (4/6 = 66.67%), D => C (4/6 = 66.67%)
	AB => E (4/5 = 80.0%), E => AB (4/5 = 80.0%)
	AE => B (4/4 = 100%), BE => A (4/5 =80%)
	A => BE (4/6 = 66.67%),

CE (1/2=0.5)	E => C (3/5 = 60%)

Figure 3 : Association rules (minconf= 60%)

Further, if the set of all the maximal frequent itemsets of a transaction database are discovered with respect to given value of minsup (σ), then the set of all the frequent sets can be found without any extra scan of the database and thus the set of all maximal frequent itemsets gives a compact representation of the set of all the fequent itemsets [12].

4. RELATED WORKS

There were some related but not directly applicable works, when association rule mining was started as a research area. This included the induction of classification rules, discovery of cusal rules, learning of logical definitions, fitting of functions to data and clustering. The closest work in the machine learning literature is the KID3 algorithm. If used for finding all association rules, this algorithm needs as many passes over the data as the number of combinations of items in the antecedent, which is exponentially large. Related work in the database literature is the work on infereing functional dependencies from data. [8] **[13] [14] [15]**.

Initial works on the quantification of the interestingness of association rules were mostly attributed to piatesky – Shapiro. For mining association rules several algorithms are proposed [3] [7] [8] [16] [17] [18] [19] [20] [21] [22] [23]. The apriori algorithm [7] [8] is one of the leading algorithms for mining association rules. This algorithm proceeds in levelwise manner in the itemset lattice and employs an efficient candidate generation technique in which only the frequent itemsets found at a level are used to construct candidate itemsets for the next level. But the algorithm needs multiple passes equal to the size of the longest frequent itemset. DHP [21] algorithm which uses hash based approach also requires many database passes equal to the size of the longest itemset. A. Savasere, E. Omiecinsky, and S. Navathe proposed the partition algorithm in [22] to minimise the database scans to only ***two*** by partitioning the database into small partitions so that each partition can be accommodated in the main memory. However, computational overhead increases in this algorithm. DLG [24] is another algorithm which assumes memory resident bit vectors for each itemset containing the TIDs of the occurrence of the itemset and then frequent itemsets are generated by logical AND operations on the bit vectors. But the lengths of the bit vectors are proportional to the number of transactions in the database and thus it may grow too long to be accommodated in the main memory for large number of transactions. A method of counting candidate sets of different length dynamically with the progress of the database scans called Dynamic Itemset Counting (DIC) is proposed by Brin, Motwani, Ullman, and Tsur in [16]. Sampling method was also used with a view to reduce I/O overhead [23]. AS – CPA proposed by Lin and Dunham [19] is another algorithm based on partition which needs at most two scans of the database. In [17] and [26] approaches using only general purpose DBMS systems and relational algebra operations are studied in the context of discovery of association rules. Pincer – Search [18] and All MFS [28] are two algorithms for discovering maximal frequent itemsets. All MFS uses randomized approach while in the Pincer – Search a combination of the bottom up approach like the a priori method and a top down search with a view to reduce the database scans is employed. MaxMiner [29] is another algorithm to discover the maximal elements by narrowing the search space using efficient prunning technique. In all these research works it is found that the task of enumerating all the frequent itemsets is computationally challenging and therefore attention is shifted towards designing parallel algorithms for association rule mining. [3] [30] [31] [32]. The possibility of Integration of Association Rule Mining with Relational Database Management Systems were studied and the benefits of using vertical database layout is also discussed in [27].

Apart from the Apriori-based approach other scalable frequent itemset mining methods like FP-growth was proposed by Han, Pei, and Yin [33]. It is a pattern-growth approach for mining frequent itemsets without candidate generation. An array-based implementation of prefix-tree-structure for efficient pattern growth mining was proposed by Grahne and Zhu [34]. ECLAT, an approach for mining frequent itemsets by exploring the vertical data format, was proposed by Zaki [02].

Method for mining strong gradient relationships among itemsets was proposed in [35]. Comparative studies of different interestingness measures were done in [36] and [37]. The use of *all confidence* as a correlation measure for generating interesting association rules was undertaken in [38] and [39].

By mining compressed sets of frequent patterns attempts are made to reduce the huge set of frequent patterns generated in data mining in recent studies. Mining closed patterns can be viewed as lossless compression of frequent patterns. Lossy compression of patterns includes studies of maximal patterns in [40] and top-*k* patterns in [41]. In [42], it is proposed to use *k* – itemsets to cover a collection of frequent itemsets. A profile based approach is proposed in [43] and in [44] a clustering-based approach is proposed by Xin, Han, Yan, and Cheng for frequent itemset compression.

Use of Bayesian networks to identify subjectively interesting patterns is discussed in [45]. In various recent works [45] [46] [47] [48] [49] [50] [51] [52] [53] [54] [55] [56] interestingness of frequent itemsets, support confidence framework, sequential pattern mining, weighted association rules, semantic knowledge mining, hardware based approach for association rule mining, mining closed item sets and selective mining of association rules are carried out. A discussion on various quality measures of data mining are given in [56] and [57]. In [58] a post processing technique for association rule with interactive pruning and filtering of discovered rules using ontologies is proposed in order to reduce the number of rules. Further, a rule Schema formalism is introduced for user expectations.

5. MINING ASSOCIATION RULES WITH INTER ITEMSET DISTANCE

A number of algorithms have been developed for association rule mining with various measures of interestingness among which the support and confidence being the most common. A limitation of this approach for generating frequent itemsets is it does not take into account the relative separation of occurrences of an itemset in the transaction database. This relative separation or gap can be expressed in terms of the number of intervenning transactions in which the itemset does not occur between its two successive occurrences. This is called inter itemset distance and it is measured within the *lifespan of occurrence of an itemset* in the database. For an itemset with solitary occurrence in the whole database, inter itemset distance cannot be defined. The minimum value of inter itemset distance of an itemset is zero if the itemset has occurred in two consecutive transactions and otherwise it is non zero. Since the occurrence pattern of an itemset in the transactions of the database is random therefore the values of inter itemset distance of itemset will be diferrent between every pair of occurrence of the itemset. Therefore, an average of all the inter itemset distances between every pair of occurrences of each itemset is calculated. Thus, an item in the transaction database is associated with an average value of inter itemset distance which is a real number. The value of inter itemset distance (IID) can be different for two itemsets having the same value of support. The value of inter itemset distance expresses quantitatively the pattern of distribution of an itemset across the transaction database. Here in this work, this aspect is incorporated for mining frequent itemsets and dicovery of association rules with their average inter itemset distances based on some prespecified threshold of average inter itemset distance and support and the corresponding association rules with prespecified confidence, support and average inter itemset distance. The *total inter itemset distance of an itemset* is the sum of the lengths of all the gaps of occurrences of the itemset within the lifespan of the itemset in the transaction database. *The average inter itemset distance of an itemset is calculated by dividing the total inter itemset distance of an itemset by the number of gaps that occurred with the occurrence of the itemset in its lifespan in the database.* The length of a gap is zero when the itemset occurs in two consecutive transactions of the database.

Value of average inter itemset distance smaller than the threshold for a frequent itemset means that the itemset has occurred in the database more closer to each other and thus it indicates relative average concentration of frequent itemsets in the transaction database. In this manner, it is proposed average inter frequent 1 – itemset distance for a fequent itemset of cardinality 1 and in general, average inter frequent n – itemset distance for a fequent itemset of cardinality n. Therefore, average inter frequent n – itemset distance for a frequent n – itemset indicates the average number of transactions separating each occurrence of the itemset in the database. In figure 2(a) calculation of average inter frequent n – itemset distances for each frequent n – itemset of the eaxmple database is shown. Thus, for instance, if an item occurs in three consecutive transactions then there are two gaps each one of length zero between the first and the second occurrence and between the second and the third occurrence of the itemset respectively. ***An itemset is called closely spaced frequent n –itemset or closely spaced large n –itemset if it is a Closely spaced n – itemset and a frequent n – itemset at the same time based on some prespecified threshold values.*** The average inter itemset distance is a measure of nearness of occurrences of an itemset in the transactions from each other in its lifespan.

For a Closely Spaced Frequent or Large n – Itemset X

$$\text{Supp}(X) \geq \sigma \text{ and Average IID } (X) \text{ or Spread } (X) \leq d$$

Where, σ is the threshold for support and d is the threshold value for maximum average IID or spread for the itemset X which is specified as input. The ranges of **σ** and **d** are such that

σ: [0, 100], σ is the percentage support
d: [0, (n – nσ)/(nσ – 1)], n = | D| i.e. number of transactions in D.

Discovering all closely spaced itemsets along with their average inter itemset distances and all the closely spaced frequent itemsets along with their supports and average inter itemset distance is a non trivial problem if the cardinality of I, the set of all the items of the database of transactions D is large. The problem is to identify which of the subsets of I are frequent and closely spaced. Average Inter Itemset Distance (IID) or Spread of an itemset is defined as the average separation of the occurrences of the same itemset in its lifespan in a database of transaction. It is given as

Average Inter Itemset Distance or Spread (d)
= (Sum of the lengths of all the gaps of occurrences of an itemset within its lifespan in terms of the number of transactions of non occurrence) / (Support of the itemset – 1).

$$\mathbf{d = (\textstyle\sum^{m-1}{}_{1=1}\ d_{i,\,i+1}) / (s - 1)} \qquad \mathbf{(3)}$$

Where, m is the TID of the transaction in which the itemset has its last appearance. The apriori algorithm is modified for the calculation of d and $d_{i,\ i+1}$. Initially, s = 0 and $d_{i,\ i+1} = 0$. Whenever, the occurrence of an itemset is detected s is incremented by 1 otherwise the value of $d_{i,\ i+1}$ is incremented by 1 but this incrementing of the value of $d_{i,\ i+1}$ starts only after the first occurrence of an itemset in the transactions. This is continued untill the transaction of last occurrence of the itemset is encountered. The value of average inter itemset distance or spread of an itemset cannot be calculated for an itemset with value of support count just 1 because it does not produce any gap with its subsequent occurrences in the transactions. The same is true for itemsets with support count zero, since the itemset has not occurred at all. Therefore in such cases, the value of average inter itemset distance or spread of an itemset cannot be defined. Further, an itemset having high value of support with respect to input support threshold need not necessarily have low value of average inter itemset distance. The lifespan of occurrence of an itemset which is frequent and closely spaced is found without requiring to make any additional scan of the database. It is also observed that itemsets with same support and same size does not have the same value of average inter itemset distance or spread always.

6. MINING ASSOCIATION RULES WITH AVERAGE INTER ITEMSET DISTANCE

Mining association rules with average inter itemset distance, support and confidence refines the association rules discovered with support and confidence. Average inter itemset distance is introduced as another measure of interestingness for the association rules in this work. An algorithm is designed based on the levelwise approach of the standard apriori algorithm and is described below. We call the association rules which satisfy the prespecified values of support, confidence and average inter itemset distance as the ***closely spaced association rules*** to distinguish them from the conventional association rules.

6.1 Problem Decomposition

The problem of mining association rules with average inter itemset distance, support and confidence can be decomposed into three broad steps as

(i) **Step 1:** Find all the itemsets having support greater than or equal to the user specified minimum support threshold σ.

(ii) **Step 2:** Find the average inter itemset distance or spread (d) for each of the frequent itemsets discovered in step 1.

The actions of these two steps are performed in the same loop and in the same pass of the algorithm for each scan of the database. This process is continueed till all the frequent n – itemsets and all the closely spaced frequent n – itemsets are discovered. Like apriori, this also takes n scans over the database. Then the frequent n – itemsets and closely spaced n – itemsets are stored along with their support and average inter itemset distance.

(iii) **Step 3:** Use the frequent and closely spaced itemsets to generate the association rules with respect to the prespecified threshold values. We call such association rules as **closely spaced association rules.**

6.2 Proposed Algorithm:

Based on the above problem decomposition an algorithm is proposed in line with the apriori method. The proposed algorithm has the following segments:

(i) Mining Closely Spaced Large – 1 Itemsets (SL_1)

Computes the Large 1-Itemsets (L_1) and Closely Spaced 1-Itemsets (S_1) and then compute Closely Spaced Large 1 – Itemsets ($CSLI_1$) by the intersection $L_1 \cap S_1$.

(ii) Mining Closely Spaced Large – k Itemsets (SL_k).

(iii) Generating Candidate k – Itemsets (SC_k) from the large (k – 1) - itemsets (L_{k-1}) discovered in every previous pass using the function Generate Candidate Itemsets (L_{k-1}).

(iv) Prune Candidate k – Itemsets (SC_k)

(v) Mining Closely Spaced Large Itemsets (SL) of all the sizes.

(vi) Generate closely spaced association rules from closely spaced large Itemsets (SL).

6.3 Analysis of the Algorithm:

The computational complexity of the proposed modified algorithm depends upon the following factors.

(i) Support Threshold: The less the support threshold the more is the number of frequent itemsets. This adverseley affects the computational complexity of the algorithm because this leads to generation and counting of more candidate sets.

(ii) Average Inter Itemset Distance Threshold: The value of the Average Inter Itemset Distance threshold will not affect the computational complexity of the proposed algorithm in the sense that it does not require to make any extra pass of the dataset while counting the value of the Average Inter Itemset Distance of each candidate itemset. The significance of Average Inter Itemset Distance is: the lesser the value of Average Inter Itemset Distance of an itemset, the more nearer are the occurrences of the itemset in the transactions of the dataset. As a result, the number of qualified itemsets for rule generation will reduce and hence the number of generated rules will also reduce. The quantiy of reduction in the number of frequent itemsets as compared to the conventional apriori approach will depend on the threshold values of both the parameters.

(iii) Number of Items: The space requirement to store the support counts and the Average Inter Itemset Distances of the itemsets increases with the increase of the items in the dataset since the number of candidate and the frequent itemsets grows in this case. As the number of frequent itemset increases with the dimentionality of the data this leads to generation of more number of candidate itemsets. Under such circumstances the computation and the I/O cost increases.

(iv) Number of Transactions: If the the number of transactions increases, then the run time increases since in the apriori based approaches the algorithm needs to make multiple passes over the dataset.

(v) Average Transaction Width: With the increase in the average transaction width, the maximum size of the frequent itemsets likely to increase. This causes generation of more candidate itemsets and more number of database passes.

(vi) Time Complexity of the Proposed Algorithm:

(a) Generation of Frequent -1 and Closely Spaced – 1 Itemsets: These two tasks are performed in the same loop of the algorithm and hence no extra scan of the database is required to calculate the Average Inter Itemset Distances of the **Candidate – 1** itemsets. We know that every item present in a transcation is a Candidate – 1 itemset for being frequent – 1 and closely spaced – 1 itemset. Therefore, the support count and the Average Inter Itemset Distance of each item present in a transaction is calculated in this step and then the itemsets are subjected to the specified threshold conditions for determining the Frequent -1 and Closely Spaced – 1 Itemsets denoted by L_1 and S_1 respectively. Thereafter, the set of **Closely Spaced Frequent -1 Itemsets** are found by the intersection of L_1 and S_1. If w is the average transaction width and **n** is the total number of transactions in the database then this operation requires O(nw) time.

(b) Candidate Generation: To generate candidate k–itemsets, pairs of frequent (k – 1)–itemsets are merged to determine whether they have at least k – 2 common elements. Each merging operations requires at most k – 2 equality comparisons. In the best case, every merging step produces a viable candidate k–itemset. In the worst case, the algorithm merges every pair of frequent (k – 1)–itemsets found in the previous iteration. Therefore, the overall cost of merging frequent itemsets is

$$\sum^{w}_{k=2} (k-2)\,|C_k| < \text{Cost of Merging} < \sum^{w}_{k=2} (k-2)|F_{k-1}|^2$$

During candidate generation a hash tree is also constructed to store the candidate itemsets. The cost for populating the hash tree with candidate itemsets is $O(\sum^{w}_{k=2} k\ |C_k|)$, where k is the maximum depth of the tree.During candidate pruning, we need to verify that the (k – 2) subsets of every candidate k – itemset are frequent. Since the cost for looking up a candidate in a hash tree is O(k), the candidate pruning step requires $O(\sum^{w}_{k=2} k(k-2)\,|C_k|)$ time.

(c) Support Counting: The number of itemsets of size k produced by a transaction of length length |t| is ${}^{|t|}C_k$ and the number of hash tree traversals required for each transaction is also equal to ${}^{|t|}C_k$. If w is the maximum transaction width and σ_k is the cost of updating the support count of a candidate k – itemset in the hash tree, then the cost of support counting is $O(N\sum_k ({}^{w}C_k\ \sigma_k))$. Since, for counting the Average Inter Itemset Distances of the itemstes no additional loop is employed and it is done in the same loop used for support counting, therefore the cost of calculating the Average Inter Itemset Distances of the itemstes is $O(N\sum_k ({}^{w}C_k\ d_k))$, where d_k is the cost of updating the Average Inter Itemset Distance of a candidate k – itemset in the hash tree. Therefore, the total cost of support counting and calculating the Average Inter Itemset Distances is $O(N\sum_k ({}^{w}C_k\ (\sigma_k + d_k)))$.

(d) Rule Generation: A closely spaced large k – itemset can produce upto $(2^k - 2)$ association rules excluding the rules which have empty antecedents (Φ=>Y) and empty consequents (Y=> Φ). The calculation of confidence of a **closely spaced association rule** does not require additional scans of the transaction database since it can be calculated by using the supports of the itemsets (X U Y) and X of the rule X=>Y in the ratio *sup(X U Y)/sup(X)*.

7. WORKING OF THE ALGORITHM

Below the working of the modified algorithm presented above is described with an example database D given in fig 4. Let us assume the prespecified input threshold values for **support to be 20% (i.e. 04)** and for the **Average Inter Itemset Distance to be 3.0.**

TID	**Items**								
001	A		C			F		H	I
002	A	B	C				G		
003		B	C	D			G		I
004	A		C						I
005			C	D	E			H	I
006	A			D		F	G		
007	A	B	C				G		
008	A		C						I
009			C	D	E			H	I
010		B	C	D			G		I
011	A		C			F		H	I
012	A			D		F	G		
013		B		D			G		I
014		B	C	D			G		I
015				D	E	F	G		
016	A	B	C		E				
017					E	F		H	I
018				D		F		H	
019		B	C	D			G		I
020	A					F			I

Figure 4: Database of Transactions

(i) **Database Scan No. 1:**

(a) Generating Candidate 1 – Itemsets (SC_1), Large 1 – Itemsets (L_1), Closely Spaced 1 – Itemsets (S_1) and the Closely Spaced Large 1 – Itemsets (SL_1).

Itemset	Support	Average IID	L_1 (20% =4)	S_1 (3.0)	$SL_1 = S_1 \cap L_1$
A	10	10/9 = 1.1	A	A	A
B	8	10/7 = 1.4	B	B	B
C	13	6/12 = 0.5	C	C	C
D	11	6/10 = 0.6	D	D	D
E	5	8/4 =2.0	E	E	E
F	8	12/7 = 1.7	F	F	F
G	10	8/9 = 0.9	G	G	G
H	6	12/5 = 2.4	H	H	H
I	13	7/12 = 0.6	I	I	I

Table 1: Results of Database Scan No 1

Now, we get $|L_1| =9$, $|S_1| =9$ and $|SL_1| =9$

(b) **Generating Candidate Itemsets (SC_2) from L_1:**
SC_2 = {AB, AC, AD, AE, AF, AG, AH, AI, BC, BD, BE, BF, BG, BH, BI, CD, CE, CF, CG, CH, CI, DE, DF, DG, DH, DI, EF, EG, EH, EI, FG, FH, FI, GH, GI, HI}

(c) **Pruning the Candidate Itemsets (SC_2) Generated from L_1:**
Prune(SC_2) = {AB, AC, AD, AE, AF, AG, AH, AI, BC, BD, BE, BF, BG, BH, BI, CD, CE, CF, CG, CH, CI, DE, DF, DG, DH, DI, EF, EG, EH, EI, FG, FH, FI, GH, GI, HI}

(ii) **Database Scan No. 2:**
(a) Genarating Large 2 – Itemsets (L_2), Closely Spaced 2 – Itemsets (S_2) and the Closely Spaced Large 2 – Itemsets (SL_2).

Itemset	Support	Average IID	L_2 (20% =4)	S_2 (3.0)	$SL_2 = S_2 \cap L_2$
AB	3	12/2 =6.0	… … …	… … …	… … …
AC	7	9/6= 1.5	AC	AC	AC
AD	2	5/1=5.0	… … …	… … …	… … …
AE	1	… … …	… … …	… … …	… … …
AF	5	15/4=3.8	AF	… … …	… … …
AG	4	7/3=2.3	AG	AG	AG
AH	2	9/1=9.0	… … …	… … …	… … …
AI	5	15/4=3.8	AI	… … …	… … …
BC	7	11/6=1.8	BC	BC	BC
BD	5	12/4=3.0	BD	BD	BD
BE	1	… … …	… … …	… … …	… … …
BF	0	… … …	… … …	… … …	… … …
BG	7	11/6=1.8	BG	BG	BG
BH	0	… … …	… … …	… … …	… … …
BI	5	12/4=3.0	BI	BI	BI
CD	6	11/5=2.2	CD	CD	CD
CE	3	9/2=4.5	… … …	… … …	… … …
CF	2	9/1=9.0	… … …	… … …	… … …
CG	6	12/5=2.4	CG	CG	CG
CH	4	7/3=2.3	CH	CH	CH
CI	10	9/9=1.0	CI	CI	CI
DE	3	8/2=4.0	… … …	… … …	… … …
DF	4	9/3=3.0	DF	DF	DF

DG	8	9/7=1.3	DG	DG	DG
DH	3	11/2=5.5	… … …	… … …	… … …
DI	7	10/6=1.7	DI	DI	DI
EF	2	1/1=1.0	… … …	EF	… … …
EG	1	… … …	… … …	… … …	… … …
EH	3	10/2=5.0	… … …	… … …	… … …
EI	3	10/2=5.0	… … …	… … …	… … …
FG	3	7/2=3.5	… … …	… … …	… … …
FH	4	14/3=4.7	FH	… … …	… … …
FI	4	16/3=5.3	FI	… … …	… … …
GH	0	… … …	… … …	… … …	… … …
GI	5	12/4=3.0	GI	GI	GI
HI	5	12/4=3.0	HI	HI	HI

Table 2: Results of Database Scan No 2

Now, we get $|L_2|$ =19, $|S_2|$ =16 and $|SL_2|$ =15

(b) Candidate Generation: Generating Candidate Itemsets SC_3 from L_2.

SC_3 = {**ABI, ACF**, ACG, ABC, ACD, ACH, ACI, ABG, **ADF, ADI,** ADG, **AFG**, **AFI, AFH,** AGI, **AHI,** BCD, BCG, BCI, BCH, BDG,BDI, BDF, **BFI,** BGI, BHI, CDG, CDH, CDI, CDF, **CFH, CFI,** CGH, CGI, CHI, DFG, **DFH,** DFI, DGI, DHI, **FGI, FHI,** GHI}

(c) **Prune(SC_3)**: After pruning the Candidate Itemsets (SC_3) Generated from L_2 we get

SC_3 = {ACG, ACI, **AFI,** AGI, BCD, BCG, BCI, BDG, BDI, BGI, CDG, CDI, CGI, CHI, DFI, DGI, **FHI**}

(iii) **Database Scan No. 3:**

(a) Genarating Large 3 – Itemsets (L_3), Closely Spaced 3 – Itemsets (S_3) and the Closely Spaced Large 3 – Itemsets (SL_3).

Itemset	Support	Average IID	L_3 (20% =4)	S_3 (3.0)	$SL_3 = S_3 \cap L_3$
ACG	2	4/1=4.0	…. …. ……	…. …. ……	…. …. ……
ACI	4	7/3=2.3	ACI	ACI	ACI
AGI	0	…. …. ……	…. …. ……	…. …. ……	…. …. ……
AFI	3	17/2=8.5	…. …. ……	…. …. ……	…. …. ……
BCD	4	13/3=4.3	BCD	…. ….	…. …. ……

				……	
BCG	6	12/5=2.4	BCG	BCG	BCG
BCI	4	13/3=4.3	BCI	…. …. ……	…. …. ……
BDG	5	12/4=3	BDG	BDG	BDG
BDI	5	12/4=3	BDI	BDI	BDI
BGI	5	12/4=3	BGI	BGI	BGI
CDG	4	13/3=4.3	CDG	…. …. ……	…. …. ……
CDI	6	11/5=2.2	CDI	CDI	CDI
CGI	4	13/3=4.3	CGI	…. …. ……	…. …. ……
CHI	4	7/3=2.3	CHI	CHI	CHI
DFI	0	…. …. ……	…. …. ……	…. …. ……	…. …. ……
DGI	5	12/4=3	DGI	DGI	DGI
FHI	3	14/2=7	…. …. ……	…. …. ……	…. …. ……

Table 3: Results of Database Scan No 3

Now, we get $|L_3| = 12$, $|S_3| = 8$ and $|SL_3| = 8$

(b) Candidate Generation: Generating Candidate Itemsets SC_4 from L_3.

SC_4 = {**ABCI, ACDI**, ACGI, ACHI, BCDG, BCDI, BCGI, BCHI, BDGI, CDGI, CDHI, CGHI}

(c) **Prune(SC_4)**: After pruning the Candidate Itemsets (SC_4) Generated from L_3 we get

SC_4 = {BCDG, BCDI, BCGI, BDGI, CDGI}

(iv) **Database Scan No. 4:**

(a) Genarating Large 4 – Itemsets (L_4), Closely Spaced 4 – Itemsets (S_4) and the Closely Spaced Large 4 – Itemsets (SL_4).

Itemset	Support	Average IID	L_4 (20% =4)	S_4 (3.0)	$SL_4 = S_4 \cap L_4$
BCDG	4	17/3=5.7	BCDG	…. ….	…. ….
BCDI	4	17/3=5.7	BCDI	…. ….	…. ….
BCGI	4	17/3=5.7	BCGI	…. ….	…. ….
BDGI	5	17/4=4.3	BDGI	…. ….	…. ….
CDGI	4	17/3=5.7	CDGI	…. ….	…. ….

Table 4: Results of Database Scan No 4

Now, we get $|L_4| = 5$, $|S_4| = 0$ and $|SL_4| = 0$

(b) Candidate Generation: Generating Candidate Itemsets SC_5 from L_4.

SC_5 = {BCDGI}

(c) **Prune(SC_5)**: After **puning the Candidate Itemsets (SC_5) Generated from L_4 w**e get

SC_5 = {BCDGI}

(v) **Database Scan No. 5:**

(a) Genarating Large 5 – Itemsets (L_5), Closely Spaced 5 – Itemsets (S_5) and the Closely Spaced Large 5 – Itemsets (SL_5).

Itemset	Support	Average IID	L_5 (20% =4)	S_5 (3.0)	$SL_5 = S_5 \cap L_5$
BCDGI	4	17/3=5.7	BCDGI	…. ….	…. …. ……

Table 5: Results of Database Scan No 5

Now, we get $|L_5|$ =1, $|S_5|$ =0 and $|SL_5|$ =0. No further candidate generation is now possible and hence the algorithm stops here. The details of the finding of the algorithm are as below:

(i) The set L of all the large itemsets generated with respect to the specified input thresholds is

$L = L_1 \cup L_2 \cup L_3 \cup L_4 \cup L_5$ and $|L|$ =46

(ii) The set S of all the Closely Spaced Itemsets generated with respect to the specified input thresholds is

$S = S_1 \cup S_2 \cup S_3 \cup S_4 \cup S_5$ and $|S|$ =33

(iii) The set SL of all the Closely Spaced Large Itemsets generated with respect to the specified input thresholds is

$SL = SL_1 \cup SL_2 \cup SL_3 \cup SL_4 \cup SL_5$ and $|SL|$ =32

Result Summary: With reference to the input threshold support =20% and average inter itemset distance =3.0 the following are the summary of the results:

(i) Size of the largest itemset discovered and Number of database scans performed = 5

(ii) Total number of large itemset discovered = 46

(iii) Total number of closely spaced itemset discovered = 33

(iv) Total number of closely spaced large itemset discovered = 32

Thus we have found that the number of closely spaced large itemsets discovered is reduced by 30.4% as compared to the number of large itemsets when average inter itemset distance is applied as a measure of interestingness for the discovery of large itemsets. Accordingly, the number of closely spaced association rules generated will also be reduced as compared to the conventional association rules. This approach thus brings out a way to reduce the number of association rules discovered by reducing the number of frequent itemsets discovered with the help of additional constraint of average inter itemset distance.

8. Implementation And Results

The conventional and the modified apriori algorithms are implemented in Java with windows XP operating syatem in a PC with Intel Core2 Duo Processor and 512MB of RAM. The data set used is the *retail* dataset of size 4.2MB available in the UCI repositories. As proposed and as seen from above it is found that the number of discovered large itemsets and hence the corresponding association rules in the modified version of the algorithm is considerably reduced with the introduction of the average inter itemset distance as a new measure of interestingness. We are calling such rules as the closely spaced association rules as these are discovered from the closely spaced large itemsets. The results of the implementation are shown in tables 6 and 7 below.

Size	No. of Transactions	min_sup count	No. of Large Itemsets	No. of Rules	Execution Time (In Secs)
50 kB	1329	26	64	66	48.5
100 kB	2381	47	57	51	104.266
150 kB	3285	65	69	62	234.469
200 kB	4398	87	69	61	340.188
250 kB	5469	109	62	55	452.641
300 kB	6469	129	63	56	578.594
350 kB	7634	152	65	56	692.469
400 kB	8815	176	63	52	824.422
450 kB	9818	196	66	56	974.844
500 kB	11055	221	65	58	1079.828

Table6: Results of implementation of Apriori Algorithm (min_ sup (2%) and min_conf (40%))

Size	No. of Transactions	min_sup count	No. of Large Itemsets	No. of Rules	Execution Time (In Secs)
50 kB	1329	26	17	15	53.453
100 kB	2381	47	16	15	143.922
150 kB	3285	65	16	17	269.5
200 kB	4398	87	15	14	365.343
250 kB	5469	109	15	13	491.703
300 kB	6469	129	15	15	387.719
350 kB	7634	152	14	13	522.047
400 kB	8815	176	14	11	824.187
450 kB	9818	196	14	13	953.141
500 kB	11055	221	14	13	1087.047

Table 7: Results of implementation of modified Apriori Algorithm (min_ sup (2%) and min_conf (40%)) and average IID = 15

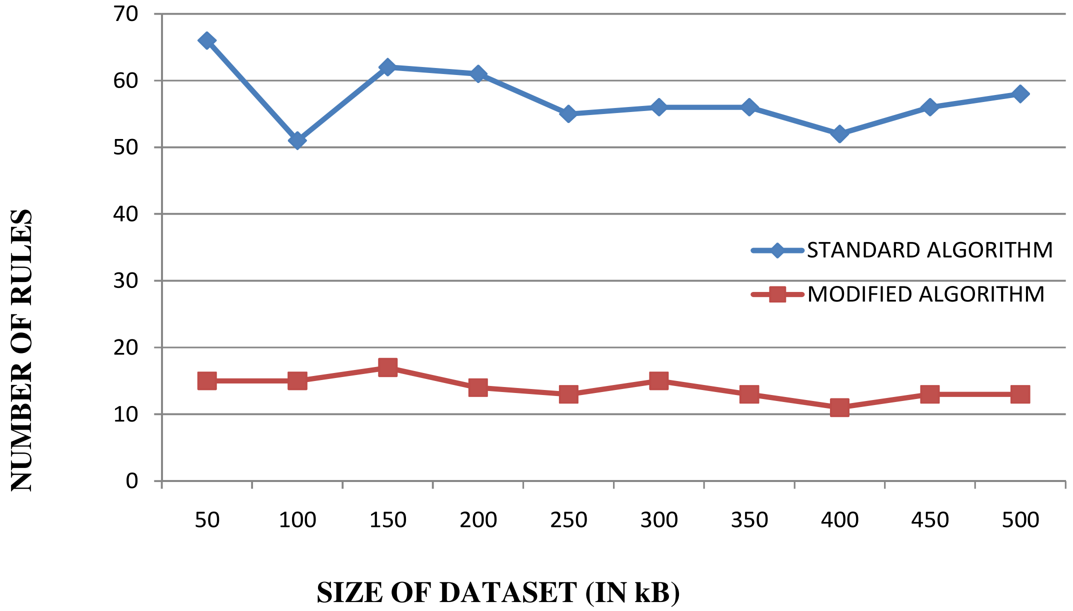

Figure 4: Graph showing the number of association rules obtained by the apriori algorithm and the modified apriori algorithm by varying the size of the dataset**.**

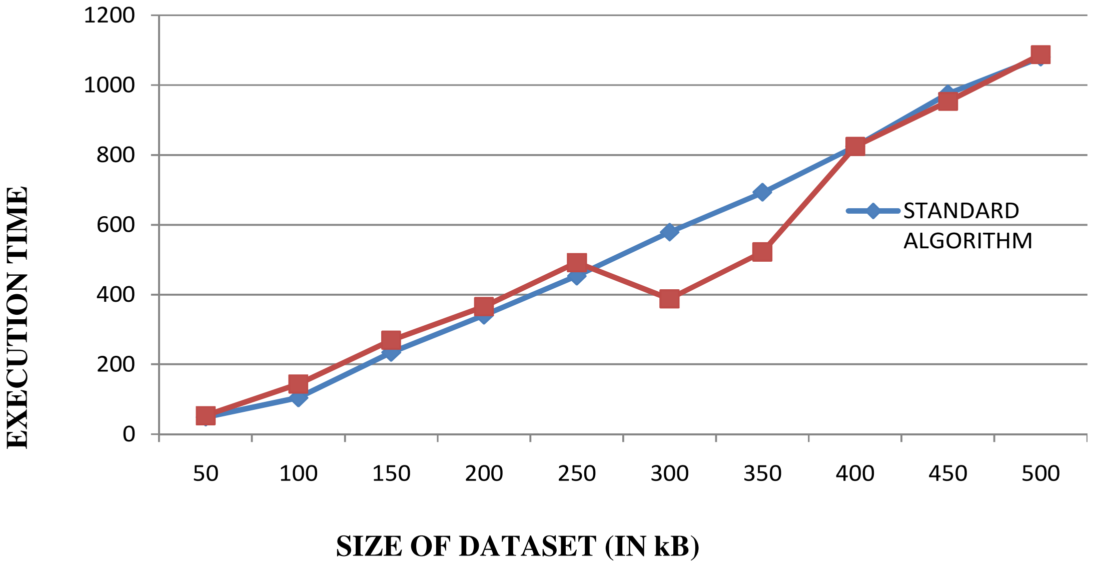

Figure 5: Graph showing the execution time required for the apriori algorithm and the modified apriori algorithm by varying the size of the dataset**.**

As shown in the graph the modified algorithm efficiently reduces the number of association rules and the execution time required are also quite comparable.

9. CONCLUSIONS AND FUTURE WORK:

Thus for the same database of transactions, the number of large itemsets and the corresponding association rules discovered with prespecified values of support and confidence is more in comparison to the number of closely spaced large itemsets and the corresponding closely spaced association rules discovered with respect to the same pair of prespecified values of support and confidence and prespecified value of average inter itemset distance. In this paper, we have introduced the concept of Inter Itemset Distance and Average Inter Itemset Distance for an itemset for the purpose of mining frequent Itemsets from database of transactions with a view to provide additional meaning to the frequent itemsets and to reduce the number of corresponding association rules which can be generated from them. Then the apriori algorithm for mining frequent itemsets and the corresponding association rules is modified and extended with the incorporation of average inter itemset distance along with support and confidence for mining closely spaced frequent itemsets and the corresponding closely spaced association rules with average inter itemset distance along with support and confidence. An experimental evaluation of the proposed algorithms shows that the number of generated rules are considerably reduced with the modified algorithm in comparioson to the conventional apriori algorithm. In the future scope the modified approach can be extended further to investigate related research frameworks of frequent itemset mining and other related topics of data mining where the notion of inter itemset distance can be found to be appropriate.

10. REFERENCE

[1] G. Piatetsky-Shapiro and W. J. Frawley, Knowledge Discovery in Databases, AAAI/MIT Press, 1991.

[2] M. J. Zaki, “Scalable Algorithms for Association Mining”, IEEE Transactions on Knowledge and Data Engineering, Vol. 12, No 3, pp 372 – 390, May/June 2000.

[3] R. Agrawal and J. Shafer, “Parallel Mining of Association Rules”, IEEE Transactions on Knowledge and Data Engineering, Vol. 8, No 6, pp 962 –969, Dec 1996.

[4] H. Toivonen, “Sampling Large Databases for Association Rules”, Proceedings of 22nd Very Large Database Conference, 1996.

[5] Sergey Brin, Rajeev Motwani, Craig Silverstein, “Beyond Market Baskets: Generalizing Association Rules to Correlations” Proceedings of the ACM International Conference on Management of Data, pages 265 -276, 1997.

[6] Margaret H. Dunham, “Data Mining: Introductory and Advanced Topics” , ISBN 81-7808 - 996 – 3, Pearson Education, 2003.

[7] R. Agrawal, T. Imielinsky, and A. Swami, “Mining Association Rules Between Sets of Items in Large Databases”, Proceedings of ACM SIGMOD Conference on Management of Data, pp207 -216, May 1993, Washington DC.

[8] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. Inkeri Verkamo “Fast Discovery of Association Rules”, Advances in Knowledge Discovery and Data Mining, U. Fayyad and et. el. Pp 307 – 328, Menlo Park, California: AAAI Press, 1996.

[9] R. Agrawal and R. Srikant, “Fast Algorithms for Mining Association Rules”, Proceedings of 20th International Conference on Very Large Databases, pp 487 – 499, Santiago, Chile, Sept 1994.

[10] G. Piatetsky-Shapiro, “Discovery, Analysis and Presentation of Strong Rules”, G. Piatetsky-Shapiro and W. J. Frawley, Editors, Knowledge Discovery in Databases, p. p. 229- 338, AAAI/MIT Press, 1991.

[11] J. Han and Y. Fu, “Discovery of Multiple Level Association Rules from Lage Databases”, Proceedings of International Conference on Very Large Databases, Zurick, pp 420 – 431, September 1995.

[12] A. K. Pujari, “Data Mining Techniques”, Universities Press, Hyderabad, India, Second edition, 2010.

[13] L. Brieman, J.H. Friedman, R.A. Olshen and C. J. Stone, “Classification and Regerssion Trees”, Belment, California, Wadsworth, 1984.

[14] P. Cheeseman, J. Kelly, M. Self, J. Stutz, W. Taylor and D. Freeman, “Autoclass: A Baysian Classification System”, Proceedings, Fifth International Conference on Machine Learning, pp 54-64, San Mateo, California, Morgan Kaufmann, 1988.

[15] D. H. Fisher, “Knowledge Acquisition via Incremental Conceptual Clustering”, Machine Learning 2(2): 139 – 172.

[16] S. Brin, R. Motowani, J. Ullman and S. Tsur, “Dynamic Itemset Counting and Iimplication Rules for Market basket Data”, ACM SIGMOD Conference on Management of Data, May 1997.

[17] M. Houtsma and A. Swami, “Set-Oriented Mining of Association Rules in Relational Databases”, 11th International Conference, Data Engineering, 1995.

[18] D. I. Lin and Z. M. Kedem, “Pincer Search: A New Algorithm for Discovering the Maximal Frequent Set”, Sixth International Conference, Extending Database Technology, March, 1998.

[19] J.L. Lin and M. H. Dunham, “Mining Association Rules: Anti Skew Algorithms,” 14th International Conference on Data Engineering, Feb. 1998.

[20] A. Mueller, “Fast Sequential and Parallel Algorithm for Association Rule Mining: A Comparison”, Technical Report CS – TR- 35/5, University of Maryland, College park, August 1995.

[21] J. S. Park, M. Chen, and P. S. Yu, “An Effective Hash Based Algorithm for Mining Association Rules”, ACM SIGMOD International Conference on Management of Data, May, 1995.

[22] A. Savasere, E. Omiecinsky, and S. Navathe, “An Efficient Algorithm for Mining Association Rules in Large Databases”, Proceedings, 21st Conference, Very Large Databases, 1995.

[23] H. Toivonen, “Sampling Large Databases for Association Rules”, Proceedings, 22nd Conference, very Large Databases, 1996.

[24] S. S. Yen and A. L. P. Chen, “An Efficient Approach to Discovering Knowledge from Large Databases”, Fourth International Conference on Parallel and Distributed Information Systems, December 1996.

[25] M. J. Zaki, S. Parthasarathi, W. Li. And M. Ogihara, “Evaluation of Sampling for Data Mining of Association Rules”, Seventh International Workshop on Research Issues on Data Engineering, August 1997.

[26] M. Holsheimer, M. Kersten, H. Mannila and H. Toivonen, “A Perspective on Databases and Data Mining”, First International Conference on Knowledge Discovery and Data Mining, August 1995.

[27] S. Sarawagi, S. Thomas, and R. Agrawal, “Integrating Association Rule Mining with Databases: Alternatives and Implications”, ACM SIGMOD International Conference on Management of Data, June 1998.

[28] D. Gunopulos, H. Mannila, and S. Saluja, “Discovering All the Most Specific Sentences by Randomized Algorithms”, International Conference on Database Theory, January 1997.

[29] R. J. Bayardo, “Efficiently Mining Long Pattrens from Databases”, ACM SIGMOD Conference on Management of Data, May 1997.

[30] D. Cheung, J. Han, V. Ng, A. Fu, and Y. Fu, “ A Fast Distributed Algorithm for Mining Association Rules”, Fourth International Conference on Parallel and Distributed Information Systems, December 1996.

[31] E. H. Han, G. Karypis, and V. Kumar,”Scalable Parallel Data Mining for Association Rules”, ACM SIGMOD Conference on Management of Data, May 1997.

[32] M. J. Zaki, S. Parthasarathi, M. Ogihara, and W. Li. “Parallel Algorithms for Fast Discovery of Association Rules”, Data Mining and Knowledge Discovery, an International Journal, Volume 1, No. 4, pp 343 – 373, December 1997.

[33] J.Han, J. Pei, and Y. Yin, “Mining Frequent Patterns without Candidate Generation”, In Proceedings of 2000 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD’00), pages 1–12, Dallas, TX, May 2000.

[34] G. Grahne and J. Zhu, “Efficiently Using Prefix-trees in Mining Frequent Itemsets. In Proceedings of ICDM’03 Int. Workshop on Frequent Itemset Mining Implementations (FIMI’03), Melbourne, FL, Nov. 2003.

[35] T. Imielinski, L. Khachiyan, and A. Abdulghani, “Cubegrades: Generalizing Association Rules”, Data Mining and Knowledge Discovery, 6:219–258, 2002.

[36] R. J. Hilderman and H. J. Hamilton, “Knowledge Discovery and Measures of Interest”, Kluwer Academic Publishers, 2001.

[37] P.-N. Tan, V. Kumar, and J. Srivastava, “Selecting the Right Interestingness Measure for Association Patterns”, Proceedings of 2002 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD’02), pages 32–41, Edmonton, Canada, July 2002.

[38] E. Omiecinski, “Alternative Interest Measures for Mining Associations. IEEE Trans. Knowledge and Data Engineering, 15:57–69, 2003.

[39] Y.-K. Lee, W.-Y. Kim, Y. D. Cai, and J. Han., “CoMine, Efficient Mining of Correlated Patterns”, Proceeding of 2003 Intternational Conference on Data Mining (ICDM’03), pages 581–584, Melbourne, FL, Nov. 2003.

[40] R. J. Bayardo, “Efficiently Mining Long Patterns from Databases”, Proceedings of 1998 ACM SIGMOD International Conference on Management of Data (SIGMOD’98), pages 85–93, Seattle, WA, June 1998.

[41] J. Wang, J. Han, Y. Lu and P. Tzvetkov, “TFP: An Efficient Algorithm for Mining Top-k Frequent Closed Itemsets”, IEEE Transaction on Knowledge and Data Engineering, 17:652–664, 2005.

[42] F. N. Afrati, A. Gionis and H. Mannila, “Approximating a Collection of Frequent Sets”’ Proceedings of 2004 ACM SIGKDD International Conference, Knowledge Discovery in Databases (KDD’04), pages 2–19, Seattle, WA, Aug. 2004.

[43] X. Yan, H. Cheng, D. Xin and J. Han, “Summarizing Itemset Patterns: A Profile-based Approach” Proceedings of 2005 ACM SIGKDD International Conference, Knowledge Discovery in Databases (KDD’05), pages 314–323, Chicago, IL, Aug, 2005.

[44] D. Xin, J. Han, X. Yan, and H. Cheng, “Mining Compressed Frequent-Pattern Sets”, Proceedings of 2005 International Conference, Very Large Data Bases (VLDB’05), pages 709–720, Trondheim, Norway, Aug. 2005.

[45] S. Jaroszewics and D. Simovici, “Interestingness of Frequent Itemsets Using Bayesian Networks and Background Knowledge”, Proceedings of the 10th International Conference on Knowledge discovery and Data Mining, pages 178 -186, Seattle, WA, August 2004.

[46]M.C. Tseng and W.Y. Lin, “Efficient Mining of Generalized Association Rules With Non-uniform Minimum Support”, Data & Knowledge Engineering 62, Science Direct, pp. 41–64, 2007.

[47] A. Ceglar and J.F. Roddick, “Association Mining”, ACM Computing Surveys, 38(2), 1-42, 2006.

[48] T. Calders and B. Goethals, “Non-derivable Itemset Mining”, Data Mining and Knowledge Discovery, 14, 171-206, 2007.

[49] F. Bodon, “A Survey on Frequent Itemset Mining”, Technical report, Budapest Univ. of Technology and Economics, 2006.

[50] Jen-Wei Huang, Chi-Yao Tseng, Jian-Chih Ou, and Ming-Syan Chen, “A General Model for Sequential Pattern Mining with a Progressive Database”, IEEE Transactions on Knowledge and Data Engineering, Vol. 20, No. 9, September 2008.

[51] Ken Sun and Fengshan Bai “Mining Weighted Association Rules without Pre Assigned Weights, “IEEE Transactions on Knowledge and Data Engineering, Vol. 20, No 4, April 2008.

[52] Rui Chang, Martin Stetter and Wilfried Brauer, “Quantitative Inference by Qualitative Semantic Knowledge Mining with Bayesian Model Averaging” , IEEE Transactions on Knowledge and Data Engineering, Vol. 20, No. 12, December, 2008.

[53] Ying-Hsiang Wen, Jen-Wei Huang and Ming-Syan Chen, “Hardware-Enhanced Association Rule Mining with Hashing and Pipelining”, IEEE Transactions on Knowledge and Data Engineering, Vol. 20, No. 6, June 2008.

[54] Zaki M.J and C. J. Hsiao, “Efficient Algorithms for Mining Closed Item sets and Their Lattice Structure,” IEEE Trans. Knowledge and Data Eng., vol. 17, no. 4, Apr. 2005.

[55] Margaret H. Dunham, “Data mining Introductory and Advanced Topics”, Pearson Education 2008.

[56] M. Hahsler, C. Buchta, and K. Hornik, “Selective Association Rule Generation,” Computational Statistic, vol. 23, no. 2, pp. 303-315, Kluwer Academic Publishers, 2008.

[57] F. Guillet and H. Hamilton, Quality Measures in Data Mining, Springer, 2007.

[58] Claudia Marinica and Fabrice Guillet, “Knowledge-Based Interactive Postmining of Association Rules using Ontologies”, IEEE Transactions on Knowledge and Data Eng., vol. 22, No. 6, pages 784 – 797, June 2010.

[59] Masakazu Seno and George Karypis, “LP Miner: An Algorithm for Finding Frequent Itemsets using Length Decreasing Support Constraint”, Proceedings of the 2001 IEEE International Conference on Data Mining, pages 505 – 512, san Jose, California, November, 2001.

Pankaj Kumar Deva Sarma

He received the B.Sc(Honours) and M. Sc. Degrees in Physics from the University of Delhi, Delhi, India before receiving the Post Graduate Diploma in Computer Application and the M. Tech degree in Computer Science from New Delhi, India. He is currently an associate professor of Computer Science in the University Department of Computer Science at the Assam University, Silchar, India. His primary research interest is in algorithms, data base systems, data mining and knowledge discovery, parallel and distributed computing, artificial intelligence and neural network. He was the former head of the department of Computer Science of Assam University and was the organizing vice president of the national conference on current trends in computer science organized at the Assam University in the year 2010.

Anjana Kakati Mahanta

She received the Bachelors and Masters Degrees in Mathematics from the Gauhati University, Guwahati, India before receiving the Ph. D. in Computer Science from the same university. She is currently a professor of Computer Science in the University Department of Computer Science at the Gauhati University, Guwahati, India. Her primary research interest is in algorithms, data base systems, data mining and knowledge discovery. She is presently the head of the department of Computer Science of Gauhati University.