Volume 2, No. 5, Sept-Oct 2011

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

Pattern Discovery using Coherent Rules

Praveen Kumar Ullengala*
Aurora Engineering College,
Bhongiri, Nalgonda A.P.India
praveenullengala@hotmail.com

Srikanth Jatla
Associate Professor and Head
Department of Computer Science and Engineering
Aurora Engineering College
Bhongiri, Nalgonda, A.P., India
jsrikanth@aurora.ac.in

Shaik Shah Nawaz
Associate Professor
Department of Computer Science & Engineering,
Aurora Engineering College,
Bhongir, Nalgonda, A.P, India

Abstract: Mining of association rules is of interest to data miners.Typically, before association rules are mined, a user needs to determine a support threshold in order to obtain only the frequent item sets. Having users to determine a support threshold attracts a number of issues. We propose an association rule mining framework that does not require a pre-set support threshold. The framework is developed based on implication of propositional logic. The experiments show that our approach is able to identify meaningful association rules within an acceptable execution time. In this paper, we present, a new Pattern mining algorithm will be proposed to discover domain knowledge report as coherent rules, where coherent rules would be discovered based on the coherent rule search algorithm.

Keywords: Associations Rule Mining, Propositional Logic, Implication, Threshold Free**.**

I. INTRODUCTION

Data mining is an Artificial Intelligence (AI) powered tool that can discover useful information within a database that can then be used to improve the action. ***Data Mining [1],*** also called as data archeology, data dredging, data harvesting, is the process of extracting hidden knowledge from large volumes of raw data and using it to make crucial business decisions. The steps in the knowledge discovery process include pre mining task as data cleaning and data integration, as well as post mining task such as pattern evaluation and knowledge representation. Many types of “interesting patterns” have been identified in the various research literatures and association rule constitute one such type. Data mining tasks to find these various pattern include characterization, discrimination, association analysis, classification and regression, cluster analysis, outlier analysis and evolution analysis.

Association Rule Mining (ARM) is a learning technique that has the advantage of discovering knowledge without the need to undergo a training process [1]. It is used to discover rules from a dataset, and each rule discovered has its importance measured against many interest measures [2] such as support and confidence.

Although ARM technique does not involve model selection, it necessitates a cut-off support threshold to be predefined to separate frequent patterns from the infrequent ones. Two item sets are said to be associated if they occur together frequently above a minimum support threshold value. There are major disadvantages to having a predefined threshold. Firstly, some rules are inevitably lost if the support threshold is set inaccurately. In addition, it is usually not possible to remove the support threshold in order to find infrequent items because ARM relies on a downward closure property of support, which necessitates a threshold to search for frequent item sets. That is, if an item set passes a minimum support requirement then all its subsets also pass this requirement. This minimum support threshold value is used as the basis for pruning, without which mining rules becomes infeasible due to the exponential search space. In summary, in traditional association rule mining, a minimum support threshold is needed, and should be determined accurately in order to produce useful rules for users.

To overcome the above limitation, we investigate the possibility of developing a new association rule mining framework that works without having to determine a support threshold. We base our framework on the notion of implication of propositional logic. We explain our proposed model in detail in section 3and 4 after a discussion of previous work is presented in section 2. Experiments based on an implementation of the framework and a discussion of the results is presented in section 5. Finally, conclusion is made in section 6.

II. PREVIOUS WORK

Recently, mining infrequent rules start to gain momentum as many have begun to accept that rules based on infrequently occurring items are also important because it represents knowledge not found infrequent rules, and these infrequent rules are often interesting [3], [4], [5]. Association among infrequent items have been relatively ignored by association mining algorithm mainly due to the problem of the large search space and the consequent explosion of total number of association rules reported.

Some of these reported rules may in fact be based on noise in the data. However, there have been some attempts towards finding infrequent association rules, such as [6], where a generalized association framework using correlation is proposed. Correlation is measured by Pearson's Goodness of Fit Chi Square measure. However, this chi-square measure suffers from the limitation of measuring the association inaccurately at small expected values, if one of the expected values is lower than the value five [6]. In practice, this is often being observed. This limit the use of a Chi Square based framework. In addition, the authors' algorithm relies on a modified support hence, is not really suitable to find infrequent rules except the ones that are above a threshold. [7] Finds independent rules measured by interest (leverage) and below a minimum support threshold. Authors in [7] also use the measure in [8], which is derived from correlation, and necessitates a minimum confidence threshold. Mining below a minimum support threshold has the same problem as mining above a maximum support threshold in the sense that the threshold needs to be accurately pre-set. In addition, the measure used in [8] inherits the drawbacks of a correlation measure in [6]. [9] Filters uninteresting rules using leverage as a measure. [10], [11] finds rules using measure such as leverage or lift; these can be performed without other thresholds in place. Since rules are found independently from a minimum support threshold, theoretically all infrequent rules may be found.

The measure of leverage, however, is non directional. A rule found using leverage does not indicate an implication that if a rule antecedent has an impact on the rule consequence vice versa. It denotes the number of co-occurrences of both antecedent and consequence item set that is above the case if both are independent to each other [12].

There is relatively little research on finding association rules that are both infrequent and interesting. Two fundamental constraints are (i) the selection of the measure used and (ii) the use of this measure to search for infrequent and interesting rule directly without post-processing the found rules. The measure should justify the search time in discovering rules. Such a measure must possess properties that can be used to search for infrequent association rules directly. Otherwise, the measure might be theoretically interesting but of limited practical use.

III. COHERENT RULES FRAMEWORK

The current section discusses the proposed theoretical Frame work for coherent rules. The salient features of the framework are, informally, (i) a novel, strong definition of association based on the notion of implication from propositional logic, (ii) the taking into account of frequency-based measures without requiring arbitrary thresholds and (iii) the use of mutually reinforcing rule pairs. These features are addressed in detail below.

We study the frequency of occurrences between two item sets and rather than relying on a minimum support threshold, we propose to compare various support values based on our definition of association.

In our study on the definition of an association, we found that association is defined in many ways of which can be referred to a number and different types of relationships among item sets. A typical definition of association is co-occurrence (1). Association can also be generalized into correlation or dependence rule [13]. Each definition has their merits. For the purpose of our model, we define association using implication of propositional logic in that an implication must be supported by its inverse. Such association rules mined has implications stronger than the typical associations based on single co-occurrences.

To illustrate our proposed framework, consider table 1 that contains relations between a rule antecedent (LHS), A and a rule consequence (RHS), C as an association rule. The rule antecedent A consists of a combination of items, called an antecedent item set X. An antecedent item set X may exist, represented by X, or absence, represented by ¬X. Similarly, the rule consequence C may contain existence or absence of consequence item set Y. They are represented as Y and ¬Y. The frequency of occurrence of X and Y is represented by Q1, X and ¬Y by Q2, ¬X and Y by Q3, finally, ¬X and ¬Y by Q4.The total of occurrence of Y is represented by C1, the Occurrence of ¬Y is given by C2, where C2 = m-C1. The same representation applied to X and ¬X with the statistics A1 and A2.

Table 1: Frequency of occurrences among antecedent and consequence item set

		A rule consequence (RHS), C		
		Y	¬Y	Total
A rule Antecedent(LHS), A	X	Q1	Q2	A1
	¬X	Q3	Q4	A2
	Total	C1	C2	m

Association rules,

a. X⇒Y is mapped to propositional logic implication $p \rightarrow q$ if and only if Q1>Q2, Q1>Q3, and Q1>Q4.
b. X⇒¬Y is mapped to propositional logic implication $p \rightarrow \neg q$ if and only if Q2>Q1, Q2>Q3, and Q2>Q4.
c. ¬X⇒Y is mapped to propositional logic Implication $\neg p \rightarrow q$ if and only if Q3>Q1, Q3>Q2, and Q3>Q4.
d. ¬X⇒¬Y is mapped to propositional logic Implication $\neg p \rightarrow \neg q$ if and only if Q4>Q1, Q4>Q2, and Q4>Q3.

Having mapped each are called pseudo implication. By pseudo implication, we mean that it approximates a real implication (according to propositional logic). It is not a real implication yet because there are fundamental differences – pseudo implication is judged true or false based on comparison of supports, which has a range of integer values. On the contrary, an implication is based on binary values. The former still depends on the frequencies of co-occurrences between item sets (supports) in a dataset, whereas the latter does not and is based on truth value. We again mapped pseudo implication into specific modes of implication called equivalents. Each equivalent would follow the same truth values of the respective relations in logic. For example, in equivalents, the negation and the inverse-negation of an implication is always false. That is, to map association rules X⇒Y to logic equivalent X≡Y, we need to check if the support value on its negation X⇒¬Y and inverse-negation ¬X⇒Y are lower than other support values.

Coherent rules are a pair of antecedent and consequence item sets, X and Y represented using a pair of rules following the truth table value for equivalents. For example, $X \Rightarrow Y$, $\neg X \Rightarrow \neg Y$, where,

a. $X \Rightarrow Y$ is mapped to logic equivalent $p \equiv q$ if and only if, Q1>Q2, Q1>Q3, Q4>Q2, and Q4>Q3.
b. $X \Rightarrow \neg Y$ is mapped to logic equivalent $p \equiv \neg q$ if and only if, Q2>Q1, Q2>Q4, Q3>Q1, and Q3>Q4. $\neg X \Rightarrow Y$ is mapped to logic equivalent $\neg p \equiv q$ if and only if, Q2>Q1, Q2>Q4, Q3>Q1, and Q3>Q4.
c. $\neg X \Rightarrow \neg Y$ is mapped to logic equivalent $\neg p \equiv \neg q$ if and only if, Q1>Q2, Q1>Q3, Q4>Q2, and Q4>Q3.(Having mapped, each rule is called pseudo implication of equivalent.)

Suppose, I= $\{i_1, i_2 \ldots i_n\}$ be a set of items. And, T= $\{t_1, t_2 \ldots t_m\}$ be a set of transaction records. A task-relevant transaction record t_j holds a subset of items such that $t_j \subseteq I$. Let I_X and I_Y be two sets of items, where, $I_X \subset I$, $I_Y \subset I$, and $I_X \cap I_Y = \emptyset$. And, let X be the antecedent item set of coherent rules, where, $X \subset I_X$ and $X \neq \emptyset$, and let Y be the consequence item set of coherent rules, where, $Y \subset I_Y$ and $Y \neq \emptyset$. Between X and Y, there are two coherent rules pairs of either,

a. $X \Rightarrow Y$, $\neg X \Rightarrow \neg Y$, and
b. $X \Rightarrow \neg Y, \neg X \Rightarrow Y$ (1)

Each coherent rules pair consists the same antecedent and consequence item set, X and Y. We called the first pair, positive coherent rules and the latter negative coherent rules because it involves absentee of an item set in each pseudo implication of equivalent.

Coherent rules are only represented using two different representations following a rule antecedent A, and a rule consequence C as follows,

a. $A \Rightarrow C$, $\neg A \Rightarrow \neg C$, and
b. $A \Rightarrow \neg C$, $\neg A \Rightarrow C$

The symbol '¬' comes from the representations, and when applied to an item set contained by A or C, it means the item is not observed in transaction records. And, since from two item sets we can write a coherent rules pair, we distinguish between coherent rules and a pair of rules that yet to be validated by calling the latter – candidate coherent rules. These can be represented differently from coherent rules using two item sets X and Y, before they are validated to be coherent rules. If the support values on these items met the binary condition of coherent rules, then they are written using one of the representations. Otherwise, they remain a pair of item sets. We use the symbol '....' and a following representation to denote this candidate coherent rules pair,

X...Y (2)

IV. COHERENT RULES SEARCH ALGORITHM

In this section, we present the internal details of the proposed algorithm to generate coherent rules. The algorithm does not require a minimum support threshold in advance. The only user-specified parameter is w, which is a percentage such that rules generated will have strength value within the top w% of the strongest strength value of coherent rules found. Typically, we are interested in a small subset of all possible rules which have the highest strength values of those that exist. We argue that nominating a desired percentage as above is much more conceptually appealing than requiring the user to nominate a support threshold. The disadvantages of pre-setting a support threshold have been highlighted in Section I.

The algorithm, called *generateNextCR*, is presented as Algorithm 1. This is a recursive algorithm that is invoked after initially setting R to null, I_Y to the complete item set except for the consequent, PV_{x1} and PV_{x2} to zero, PV_Y to the index of the consequence item set, and PV_{Max} to the cardinality of I_Y, T to the transaction records, RA to null, and a set of coherent rules found CR to null. The indexes PV_{x1} $PV_{x2}$and the buffer for indexes RA is used to refer to an antecedent item set of coherent rules. The index PV_Y refers to the index of consequence item set that is of cardinality '1'. The index PV_{Max} sets the termination criteria for the recursion, i.e. if the index PV_{x1} equals to $PV_{Max.}$ Support values are scanned from transaction records T, with coherent rules found are kept in CR.

The algorithm proceeds to systematically explore the power set of I_Y, but does not need to generate the complete Power set as that would be infeasible. The feasibility of the algorithm is ensured in two ways. Firstly, if a candidate coherent rule pair does not meet the anti-monotone properties, then coherent rules containing a superset of its item set are not generated (see Lines 4.15 – 4.15.4 in Algorithm 1). Secondly, as a logical consequence, if the cardinality of the antecedent item set of a candidate coherent rule pair that does not meet the anti-monotone property consists only of a single item, then this item can be removed from I_Y (see Lines 4.1.5.3.2 and 4.1.6.2). Clearly, such a removal cuts down the cardinality of the power set being explored by a factor of 2.

The algorithm also articulates subset of all possible coherent rules, which have the highest w% strength values within those that exist (see Lines 4.1.5.2.3 and 4.1.5.2.4). Interestingly, it does not have to calculate the strength values of all possible coherent rules in order to find the highest w% strength values. The algorithm calculates and estimates the strongest possible strength value for a group of candidate coherent rules with super sets; if they are coherent rules (see Lines 4.1.4 and 4.1.5.1). Since the strength values of coherent rules with supersets are lower than the strongest possible strength values, maxPossible_s, and maxEstPossible_s, if either one is lower than the required strength value, then we do not have to generate these candidate coherent rules. Finally, strength values are computed for those candidate coherent rules that pass the conditions (see Line 5.1.5.2.1). Based on the real strength values, the top w% of coherent rules is maintained in line 4.1.5.2.2.

A. Algorithm generateNextCR :

(candidateCoherentRules R, items I_Y, itemIndex PV_{x1}, itemIndex PV_{x2}, itemIndex PV_Y, itemIndex PV_{Max}, subItems T, ordered Set<index> RA, RuleSet CR**)**

//Initial//
1. If $PV_{x1} > 1$
1.1 PV_{x2}:= PV_{x1}, PV_{x1}:= 1
2. Else
2.1 PV_{x2}:= PV_{Max},
3. End if
//Generating candidate coherent rules by enumerating Antecedent item set X//
4. While (PV_{x1}< PV_{x2})
4.1 If (PV_{x1}! = PV_Y)
4.1.1 RA ←concatenate (PV_{x1}, RA)

4.1.2 X ←{i_L: L∈RA}
4.1.3 Let Y be the set of candidate coherent rules Corresponding to (X, Y) such that R= (X⇒Y, ¬X⇒¬Y)
//START of Conditions for Efficient Generations//
4.1.4 Compute maxEstPossible_s, Q1F, Q3F based on single scan
4.1.5 If (Q1F>Q3F) and (maxEstPossible_s ≥ min_s)
4.1.5.1 Compute maxPossible_s, Q1, Q2, Q3, Q4 using another scan
4.1.5.2 If (Q1>Q3) and (maxPossible_s ≥ min_s)
4.1.5.2.1 ∀ r ∈ R compute Hr and store it
4.1.5.2.2 Update >min_s based on user- specified _ and the strongest, [found
4.1.5.2.3 If (Hr > min_s)
4.1.5.2.3.1 CR= CR ∪ R
4.1.5.2.3.2 toRemove = {cr: cr ∈CR and Hcr <min_s}
4.1.5.2.3.3 CR = CR – toRemove
4.1.5.2.4 End
4.1.5.3 Else
4.1.5.3.1 itemToRemove = {X: X is the antecedent tem set of some r ∈R and |x| = 1}
4.1.5.3.2 I = I − itemToRemove
4.1.5.4 End
4.1.6 Else
4.1.6.1 itemToRemove = {X: X is the antecedent item set of some r ∈ R and |x|= 1}
4.1.6.2 I = I− itemToRemove
4.1.7 End
//End of Conditions for Efficient Generations//
4.1.8 If (PV_{x1} > 1)
4.1.8.1 (R, I, PV_{x1}, PV_{x2},PV_Y, PV_{Max}, RA) = generateNextCR(R, I, PV_{x1}, PV_{x2}, PV_Y, PV_{Max}, RA)
4.1.9 End
4.1.10 RA ←(RA – PV_{x1}) //remove an item from the buffer of ant. Item set//
4.2 End
4.3 PVx1 := PV_{x1}+ 1 //increase the first pointer value//
5. End
Algorithm 1: Generate Coherent Rules

V. EXPERIMENTS AND DISCUSSIONS

We have conducted a number of experiments. In this paper, we report the results of two main categories of experiment. In the first category, we want to show that our association rule mining framework can find infrequent association that may be difficult to find in traditional association rule mining. The zoo data set is used in this experiment. Our proposed framework requires less post-processing in generating the rule compared to the traditional association mining algorithm. That is, instead of finding too many rules, our algorithm finds smaller number of rules. Lastly, we measure the performance of our framework by testing its scalability. For this performance test, we created three sparse artificial datasets, and another three dense artificial datasets. In zoo dataset, we use the classes as the consequences in order to find association rules directly from data. On artificially generated datasets we use the last items as consequences.

A. Zoo dataset:

Zoo dataset [14] is a collection of animal characteristics and their categories in a zoo. This dataset is chosen because animal characteristics in each category are very well known. As a result, it is easier to verify the correctness and interestingness of rules mined. Zoo dataset contains seven categories of animals including mammal and amphibian. While mammal type of animal such as elephants, buffalos, and goats are frequently observed in this zoo, amphibian type of animal such as frog and toad are relatively rare.

We run our search algorithm without setting a minimum support threshold to obtain all rules within a window of a top 5%, and each rule contains not more than five items. We report the results as follows.

A total of 16 rules are found on mammal type of animals. All rules have strength of 1.0 out of 1.0. We verify the correctness of these rules based on known knowledge on this category of animal. For example, all mammal such as goat has no feathers but has milk and backbone therefore feathers(0), milk(1), and backbone(1) are reported associated with mammal(1). We list all rules contains not more than four items in table 2.

We found these rules describe mammal correctly. In fact, the first and the shortest rule milk ⇒ mammal describe a fundamental characteristic of a mammalian explicitly. From literature review, the second rule may be deemed redundant in comparison with the first rule because inclusion of an additional item set feathers (0), which cannot further increase the strength of rule. The strength of the first rule is already at its maximum at 1.0; any further inclusion of items may be redundant. Such a consideration however is application dependent. We could use both items, feathers (0) and milk (1) to describe mammalian more comprehensively at the same strength of 1.0. That is, an animal of mammal does not have feather but milk. If we discard feather (0), we loss this item as a descriptive.

Table 2: Rules describe mammal

Antecedent Item Set		Consequent Item Set
milk(1)	⇒	mammal(1)
feathers(0),milk(1)	⇒	mammal(1)
milk(1),backbone(1)	⇒	mammal(1)
feathers(0),milk(1),backbone(1)	⇒	mammal(1)
milk(1),breathes(1)	⇒	mammal(1)
feathers(0),milk(1),breathes(1)	⇒	mammal(1)
milk(1),backbone(1),breathes(1)	⇒	mammal(1)
milk(1),venomous(0)	⇒	mammal(1)
feathers(0),milk(1),venomous(0)	⇒	mammal(1)
milk(1),backbone(1),venomous(0)	⇒	mammal(1)
milk(1),breathes(1),venomous(0)	⇒	mammal(1)

We run the search for amphibian, and found a total of 136 rules. Again, we could not find any incorrect rules. These rules have strength 1.0. While studying at these rules, we are surprised by the fact that amphibian like frog is toothed! We confirm this via answer.com, and this is indeed correct. That is, frog in this zoo is toothed.

Comparing the two experiments, there is a large difference in their total number of occurrence in the overall transaction records. 41% of transaction records contain mammal, in comparison, only 4% of transaction records contains amphibian. That is, search for amphibian is a search for infrequent association rules, which is often missed by most association rule mining technique that demands a minimum support threshold. If we set minimum support threshold to be higher than 4% and use a typical association rule mining technique, we loss rules describing amphibian. In comparison, our technique does not necessitate a minimum support threshold, it finds all necessary rules.

On execution time wise, each running time takes less than 3 seconds on a Desktop Intel Core 2 Duo at 2 GHz and 4 GB of physical memory running Windows XP. Zoo dataset contains 101 transactions and 43 item sets. The search space on a target is $2^{2(n-1)} - (2^{(n-1)} - 1)$ where $2^{2(n-1)}$ is the total number of both positive and negative rules, and $(2^{(n-1)} - 1)$ is the total number of positive rules using a single consequence item set as a target. In this case, zoo dataset contains 2E+25 combinations of item sets. We use an optimistic assumption to grasp the size of the search space; we assume only one computation cycle time (1 / 1GHz) is needed to form and to validate a combination of item set in a single transaction. Based on this optimistic assumption, it follows that a search without pruning would require at least 6E+10 years to complete. In comparison, our search time is feasible. From these two experiments, we conclude that association rule pairs are useful to discover knowledge (both frequent and infrequent) from dataset.

B. Artificial Datasets:

We follow to generate a following three dense artificial datasets with an increase in complexity using the IBM synthetic data generator [16]. The symbols used in representing a dataset are explained below,

D: number of transactions in 000s
T: average items per transaction
N: number of items
L: number of patterns
I: average length of maxima pattern

The dense datasets have an average length of maxima pattern (I) close to average items per transaction (T), besides having a low number of patterns (L). These dense datasets have an increase number of items as follows,

a. D100T10N100L50I9,
b. D100T10N500L50I9,
c. D100T10N1000L50I9

We generate also sparse dataset with an increase in its number of items hence complexity,

a. D100T10N100L10000I4,
b. D100T10N500L10000I4,
c. D100T10N1000L10000I4

The results from experiments suggest that our search for association rule pairs is feasible within a linear or polynomial search time over an increase of complexity or items.

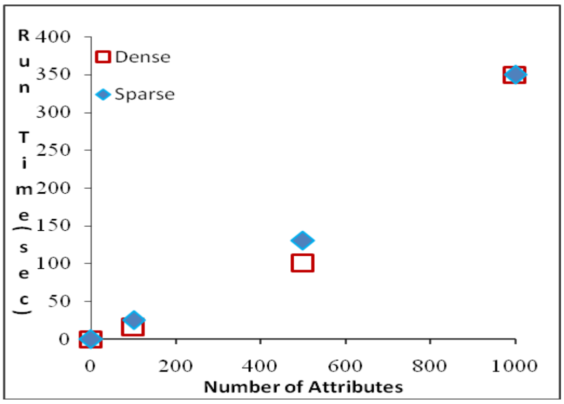

Figure 1: Search time on an increase complexity on dense and sparse dataset

VI. CONCLUSION

We have presented a framework to mine association rules without minimum support threshold. The framework employs a novel, strong definition of association based on logical equivalence from propositional logic to avoid using a cut-off support threshold. The experimental results show that implication of propositional logic is a good alternative for the definition on association.

The stronger definition of association also results in the discovery of knowledge that is vital from transaction records represented by coherent rules. These are a pair of rules that can be mapped to a pair of logical equivalents of the propositional logic, which means that the rules reinforce each other. While coherent rules found are important, the interest of these rule pairs is further quantified using coherent rules measure of interest. Coherent rules have positive values for the interest measure and imply that the antecedent item set of a coherent rule pair is needed in predicting its consequence item set, and is better than a guess without the former.

Rules based on this definition may be searched and discovered within feasible time. This can be done by our proposed strategy of finding the strongest possible strength value of a group of candidate coherent rules and comparing it to the minimum strength value required, which is constantly updated based on a parameter specified by a user. The experimental results show that it is feasible to search for coherent rules when the size of transaction records increases.

VII. REFERENCES

[1]. Agrawal, Rakesh, Imielinski, Tomasz and Swami, Arun. Mining association rules between sets of items in large databases. ACM Press, 1993, SIGMOD Rec., Vol. 22, pp. 207–216.

[2]. The Discovery of Coherent Rules, Alex Tze Hiang Sim, Maria Indrawan, Samar Zutshi, and Bala Srinivasan, IAENG International Journal of Computer Science,35:3,IJCS_35_3_18, 2008.

[3]. Liu, Bing, Hsu, Wynne and Ma, Yiming. Mining association rules with multiple minimum supports. ACM, 1999, pp. 337-341.

[4]. Cohen, Edith, et al. Finding Interesting Associations without Support Pruning. IEEE Computer Society, 2000. pp. 489-500.

[5]. Huang, Yan, et al. Mining confident co-location rules without a support threshold. ACM, 2003. pp. 497-501.

[6]. Brin, Sergey, Motwani, Rajeev and Silverstein, Craig. Beyond market baskets: generalizing association rules to correlations. ACM, 1997.Vol. 26, pp. 265-276.

[7]. Zhou, Ling and Yau, Stephen. Efficient association rule mining among both frequent and infrequent items. Pergamon Press, Inc., 2007,Comput. Math. Appl., Vol. 54, pp. 737-749.

[8]. Wu, Xindong, Zhang, Chengqi and Zhang, Shichao. *Mining Both Positive and Negative Association Rules.* San Francisco,CA,USA : Morgan Kaufmann Publishers Inc., 2002, pp. 658–665.

[9]. Piatetsky-Shapiro, G. Discovery,analysis,and presentation of strong rules. AAAI/MIT Press,Menlo Park,CA, 1991. pp. 229-248.

[10]. Zhang, Songmao and Webb, Geoffrey I. Further Pruning for Efficient Association Rule Discovery. Springer Berlin / Heidelberg, 2001, Lecture Notes in Computer Science, Vol. 2256, pp. 205-216.

[11]. Webb, G. I. and Zhang, S. k-Optimal Rule Discovery. Springer Netherlands, 2005, Data Mining and Knowledge Discovery, Vol. 10(1), pp. 39-79.

[12]. Cornelis, Mining Positive and Negative Association Rules from Large Databases. Bangkok, Thailand, June 2006. pp. 1-6.

[13]. Silverstein, Craig, Brin, Sergey and Motwani, Rajeev. Beyond Market Baskets: Generalizing Association Rules to Dependence Rules. KluwerAcademic Publishers, 1998, Data Min. Knowl. Discov., Vol. 2, pp.39–68.

[14]. AILab. Zoo dataset. [Online] January 2007. http://magix.fri.uni-lj.si/orange/doc/ datasets /zoo.htm.

[15]. Frequent Itemset Mining Dataset Repository. [Online] http://fimi.cs.helsinki.fi/data/.

[16]. IBM Quest Data Generator. [Online] July 1997. www.almaden.ibm.com/cs/quest/syndata.html.

Short Biodata of the Authors

Praveen Kumar Ullengala is pursuing his M.Tech in Software Engineering (Dept. of CSE) in Aurora Engineering College, Bhongir, Nalgonda, A.P and India. His areas of interests are data mining and knowledge Discovery, especially sequential pattern mining, Software Engineering, Testing, Network Security, Neural Networks, and Mobile computing.

Mr. Srikanth. Jatla, working as Associate Professor and Head of the Department of Computer Science and Engineering at Aurora’s Engineering College with a teaching experience of 12years.He is a B.E and M.Tech in Computer Science and pursuing his PhD in Data Stream Mining at JNTU,Hyderabad.His areas of interest includes data structures, principles of programming languages, algorithm analysis and compiler design.

Mr. Shaik Shah Nawaz has received his M.Tech degree in Computer Science. Currently, he is working as Senior Associate Professor in the Department of Computer Science & Engineering, Aurora Engineering College, Bhongir, Nalgonda, A.P, and India. He has 18-years working Experience in Software Industry. His areas of interests are Software Engineering, Testing, Network Security, Computer Networks, Wireless Communications, Data Mining and Data Warehousing.