The Discovery of Coherent Rules

Alex Tze Hiang Sim, Samar Zutshi, Maria Indrawan, Bala Srinivasan

Abstract—**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.**

Index Terms—**association rule mining, propositional logic, implication, threshold free.**

I. Introduction

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 3 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 4. Finally, conclusion is made in section 5.

Manuscript received July 24, 2008. Sim is supported by Monash PhD scholarship. Zutshi is with the Swinburne University, Melbourne, Australia. Sim, Indrawan and Srinivasan are with the Monash University, Melbourne, Australia.
(e-mail:{Tze.Sim,Maria.Indrawan,Bala.Srinivasan}@infotech.monash.edu.au, Szutshi@swin.edu.au).

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 in frequent rules, and these infrequent rules are often interesting [3], [4], [5], [6], [7]. 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 [3], [4], [5], [6], [7] , [8] , [9]. 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 [10], where a generalised 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 [10]. 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. [11] finds independent rules measured by interest (leverage) and below a minimum support threshold. Authors in [11] also use the measure in [12], 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 [12] inherits the drawbacks of a correlation measure in [10]. [13] filters uninteresting rules using leverage as a measure. [14], [15] 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 [16], [17].

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 framework 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* (10) or *dependence* rule [18]. 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* [1]. 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 [2] and a rule consequence (RHS), C [3] 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,

i) *X*⇒*Y* is mapped to propositional logic implication *p*→*q* if and only if *Q1*>*Q2*, *Q1*>*Q3*, and *Q1*>*Q4*.
ii) *X*⇒¬*Y* is mapped to propositional logic implication *p*→¬*q* if and only if *Q2*>*Q1*, *Q2*>*Q3*, and *Q2*>*Q4*.
iii) ¬*X*⇒*Y* is mapped to propositional logic implication ¬*p*→*q* if and only if *Q3*>*Q1*, *Q3*>*Q2*, and *Q3*>*Q4*.
iv) ¬*X*⇒¬*Y* is mapped to propositional logic implication ¬*p*→¬*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*⇒*Y*, ¬*X*⇒¬*Y*, where,

i) *X*⇒*Y* is mapped to logic equivalent *p*≡*q* if and only if, *Q1*>*Q2*, *Q1*>*Q3*, *Q4*>*Q2*, and *Q4*>*Q3*.
ii) *X*⇒¬*Y* is mapped to logic equivalent *p*≡¬*q* if and only if, *Q2*>*Q1*, *Q2*>*Q4*, *Q3*>*Q1*, and *Q3*>*Q4*.
¬*X*⇒*Y* is mapped to logic equivalent ¬*p*≡*q* if and only if, *Q2*>*Q1*, *Q2*>*Q4*, *Q3*>*Q1*, and *Q3*>*Q4*.
iii) ¬*X*⇒¬*Y* is mapped to logic equivalent ¬*p*≡¬*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,

i) *X*⇒*Y*, ¬*X*⇒¬*Y*, and (1)
ii) *X*⇒¬*Y*, ¬*X*⇒*Y*

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,

[1] Both inverse and contrapositive have the same total number of co-occurrences in transaction records.
[2] Non italic
[3] Non italic

i) A⇒C, ¬A⇒¬C, and
ii) A⇒¬C, ¬A⇒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,

$$X..Y \tag{2}$$

In this paper, we focus on to describe the generation of positive coherent rules.

IV. Coherent Rules Measure of interest

A. Interest Measure H based on lambda

A binary measure for coherent rules trivially follows from the definition of coherent rules in the previous section. Such a measure takes the value '1', if candidate coherent rules meet all the conditions of either coherent rules pair, and '0' otherwise. We write this below,

$$B(X..Y) = \begin{cases} 1, & Q1 > Q2, Q1 > Q3, Q4 > Q2, Q4 > Q3 \\ 1, & Q2 > Q1, Q2 > Q4, Q3 > Q1, Q3 > Q4 \\ 0, & otherwise \end{cases} \tag{3}$$

The above binary measure, however, does not differentiate the different interest of coherent rules. We propose to use an interest measure *H* based on the well-known measure of association lambda in statistics [19], [20]. Lambda measures the association of two (nominal) variables, based on a concept called Proportional Reduction in Error (PRE). According to this concept, a variable is used to predict the existence of another variable. And, if this prediction performs better than guessing the second variable independent of the first variable, then these two variables are deemed to be related to each other. Otherwise, the second variable can be guessed without the need to know the first variable. That is, the concept compares two predictions together, between knowing a variable and not knowing it. Note that we use the term *variable* in the sense of, e.g. [21] where, informally, each variable contains many categories and each category corresponds to an item set.

Lambda quantifies the strength of the association between two variables into a value between zero and one and is defined in [19], and can be rewrite as,

$$\lambda = \frac{min\binom{Q1+Q3,}{Q2+Q4} - min\binom{Q1,}{Q2} - min\binom{Q3,}{Q4}}{min\binom{Q1+Q3,}{Q2+Q4}} \tag{4}$$

For our purposes, the *Q1*, *Q2*, *Q3*, *Q4* above are computed based on the "categories" rather than the "variables" i.e. based on item sets as is traditional in data mining.

Hence, all coherent rules have an arbitrary strength value of lambda but not all associations having positive lambda value are coherent; additional conditions following propositional logic need to be met.

Based on lambda, we define the interest measure for coherent rules *H* as,

$$H(X..Y) = \begin{cases} \lambda, & B(X..Y) = 1 \\ 0, & B(X..Y) = 0 \end{cases} \tag{5}$$

It follows from the definition that lambda only takes positive value whenever subsets of the coherent rules conditions equation (3) are met. These are

i) *Q1>Q2, Q4>Q3*
ii) *Q2>Q1, Q3>Q4*

The above is demonstrated in Appendix A of [22].

B. Properties of Interest Measure H

The interest measure *H* does not inhibit anti-monotone or monotone properties. We show this below,

let *FV* be the arbitrary fixed values used in *H* such that

$$FV = \min(Q1+Q3, Q2+Q4) \tag{6}$$

and, δa, δb, δc, and δd the delta change of support values given by *Q1*, *Q2*, *Q3* and *Q4*. Suppose, the function *SV(X..Y)* finds all the support values of a candidate coherent rules pair such that

$$SV(X..Y) = (Q1, Q2, Q3, Q4) \tag{7}$$

Its support values of another candidate coherent rules having the supersets of item is,

$$SV(XE..Y) = \begin{pmatrix} Q1 - \delta a, Q2 - \delta b, \\ Q3 + \delta c, Q4 + \delta d \end{pmatrix}, \tag{8}$$

where $\delta a+\delta b=\delta c+\delta d$, and $\delta a,\delta b,\delta c,\delta d \geq 0$.

That is, the support values of *Q1* and *Q2* reduce but *Q3* and *Q4* increase over candidate coherent rules having the supersets of item.

Based on the understanding of the delta changes in support values, we now show that coherent rules measure of interest does not necessary have anti-monotone or monotone properties. The strength value of coherent rules written without its conditions is as follow,

$$H(X..Y) = \frac{FV - min\binom{Q1,}{Q2} - min\binom{Q3,}{Q4}}{FV} \tag{9}$$

If the delta changes in $\delta a,\delta b=0$, then the current strength value is at least the strength value as *H(XE..Y)*, which exhibits an anti-monotone property. That is,

$$H(X..Y) \geq H(XE..Y) = \frac{FV - min\begin{pmatrix} Q1, \\ Q2 \end{pmatrix} - min\begin{pmatrix} Q3 + \delta c, \\ Q4 + \delta d \end{pmatrix}}{FV} \tag{10}$$

If the delta changes in $\delta c,\delta d=0$, then the current strength value is at most the strength value as *H(XE..Y)*, which exhibits a monotone property. That is,

$$H(X..Y) \leq H(XE..Y) = \frac{FV - min\begin{pmatrix}Q1-\delta a,\\ Q2-\delta b\end{pmatrix} - min\begin{pmatrix}Q3,\\ Q4\end{pmatrix}}{FV} \quad (11)$$

However, *H(XE..Y)* does not necessary have both the assumptions of δa,δb=0, and δc,δd=0. Hence, it does not necessary inherits both properties. That is,

$$H(XE..Y) = \frac{FV - min\begin{pmatrix}Q1-\delta a,\\ Q2-\delta b\end{pmatrix} - min\begin{pmatrix}Q3+\delta c,\\ Q4+\delta d\end{pmatrix}}{FV} \quad (12)$$

This means, the value *H(X..Y)* can be higher, lower or the same value as *H(XE..Y)*. Hence, we cannot use the measure of coherent rules *H* to avoid generating candidate coherent rules exhaustively.

V. Mining Coherent Rules

This section covers algorithms for the generation of coherent rules. Initially we show that because *H* is defined in terms of lambda, and lambda is an interval level of measurement, it is necessary to fix the consequent item set while mining for rules.

We explain our search strategies to discover coherent rules.

A. Search Properties of Coherent Rules Based on Measure H

It is important to highlight that the coherent rules measure of interest, which is based on lambda is an interval level of measurement. It has an arbitrary zero. That is, the positive value of the measure *H(X..Y)* is given in relation to the statistics of the consequence item set. As a result, it is meaningless to compare the measure values between coherent rules that have a different consequence item set. A consequence item set must be fixed before the strength value between coherent rules that have different antecedent item sets can be compared. In this way, we make comparison within the same scale.

B. Strategy to Avoid Generating Candidate Coherent Rules Exhaustively - I

We use the statistical conditions (*Q1 > Q3*) within property of coherent rules measure of interest *H* [4], to prune supersets of item, which does not meet the condition. It follows a downward closure property of the statistical condition. Suppose, *X..Y* is a candidate coherent rules pair, and *XE..Y* is another candidate coherent rules pair such that the antecedent item set $X \subset XE$ [5]. If either of these conditions is not met by *X..Y*, then further generation of candidate coherent rules *XE..Y* is not necessary. As a result, we avoid exhaustively generate all candidate coherent rules and validate them to be coherent rules. For example, we have item sets I_X = {a, b, c, d, e}, if *X*={c, d} does not meet the condition, then further generating of candidate coherent rules with *X*={a, c, d} and *X*={b, c, d} can be avoided.

[4] Coherent rules measure in itself has no anti-monotone property.
[5] We write the union of item sets $X \cup E$ as XE.

C. Strategy to Avoid Generating Candidate Coherent Rules Exhaustively - II

In the second strategy, we design two procedures to both calculate and estimate the strongest possible strength value of the current coherent rules and a group of coherent rules with supersets of item. If these strength values are lower than a threshold value, then we can avoid generate the subsequent coherent rules with supersets of item. We highlight that the threshold value being compared to, is not provided by a user. It is determined based on a parameter provided by user. We show this in the next section, and the opportunities to avoid generate candidate coherent rules after it.

1) Minimum Strength Required

We proposed to mine arbitrary number of coherent rules with its strength value within *w%* from a yet-to-know strongest coherent rules strength value (*max_s*). That is, all coherent rules have strength value at least (1-*w*)×*max_s*. For example, a user wants to find arbitrary number of coherent rules that have strength value within top 5% of the strongest coherent rules. Let *w* = 5%, and the strongest coherent rules found from transaction records, *max_s* = 0.8. Hence, all coherent rules with strength value between 0.76 and 0.8 are reported. In another example, the user continues to find only top 5% of the strongest coherent rules, and the transaction records contain the strongest coherent rules at only *max_s* = 0.1. As a result, all coherent rules with strength value between 0.095 and 0.1 are reported. These two examples have also shown that a user does not need to understand the distributions in transaction records. They only specify the needed top *w%* from the yet-to-know strongest coherent rules.

2) Estimating The Strongest Strength Value of A Group of Coherent Rules – Part 1

This section introduces how to calculate the strongest possible strength value of coherent rules. This estimate is shown to inhibit anti-monotone property. We will use it to further avoid search space. Let, the fixed value,

$$FV = \min(Q1+Q3,\ Q2+Q4) \quad (13)$$

We re-write interest measure *H* without the conditions as,

$$H(X..Y) = (FV - \min(Q1,Q2) - \min(Q3,Q4)) / FV \quad (14)$$

H in equation (9) gives the strongest strength value of coherent rules if the right hand side holds a minimum value. For example, if *Q1* or *Q2*, and *Q3* or *Q4* holds a zero. The support values given by *Q1* and *Q2* decrease over candidate coherent rules that have the supersets of item, following anti-monotone. Hence, we could make the estimation on the strongest value of *H* over a candidate coherent rules pair and all the candidate coherent rules with the supersets of item. There are three ways to estimate the strongest strength value of coherent rules. It follows that if the strongest estimated value is lower than the arbitrary strength value required, then we could avoid generating and validating all these candidate coherent rules.

We now detail the only three ways on how these estimations are made. In each of the estimates, the estimated support values *Q2F* is assumed to be zero because following anti-monotone, candidate coherent rules that have supersets of item will have the same or a decreasing support value. By assuming *Q2F*=0, we estimate the worst support value on the candidate coherent rules with supersets of item, hence estimate the strongest possible strength value on a group of candidate coherent rules. Within a fixed size of transaction records, there are three ways on how this value can be passed. to the rest of the support values. The strongest strength value over a group of candidate coherent rules can be made by analysing the support values. These are shown below,

Assume that *Q2F=Q2-Q2*=0, and its support values is transferred to support value *Q1F=Q1+Q2*. Hence, the first possible strongest strength value, $H'_a(X..Y)$, can be calculated from support values *Q1F*, *Q2F*, *Q3*, *Q4*. (15)

Assume that *Q2F*=0, and its support values is transferred to support values *Q3F=Q3+Q2*. Hence, the second possible strongest strength value, $H'_b(X..Y)$, can be calculated from support values *Q1*, *Q2F*, *Q3F*, *Q4*. (16)

Assume that *Q2F*=0, and its support values is transferred to support value *Q4F=Q4+Q2*. Hence, the first possible strongest strength value, $H'_c(X..Y)$, can be calculated from support values *Q1*, *Q2F*, *Q3*, *Q4F*. (17)

We show the anti-monotone on the estimates ($H'_a(X..Y)$, $H'_b(X..Y)$ and $H'_c(X..Y)$) in corresponding to the above equations. The bracketed [*Q2*] is the total support value of *Q2* that is transferred into *Q3* or *Q4* hence its delta change δb=0 and this not shown. The first shows the estimate has strength value at least 'real' coherent rules strength value.

$$\begin{aligned} H'_a(X..Y) &= \frac{\begin{pmatrix} FV-min\begin{pmatrix} Q1+[Q2], \\ [Q2]-[Q2] \end{pmatrix} \\ -min\begin{pmatrix} Q3, \\ Q4 \end{pmatrix} \end{pmatrix}}{FV} \\ &= \frac{FV-min\begin{pmatrix} Q3, \\ Q4 \end{pmatrix}}{FV} \\ &\geq \frac{FV-min\begin{pmatrix} Q3+\delta c, \\ Q4+\delta d \end{pmatrix}}{FV} \\ &\geq H'_a(XE..Y) \\ &\geq \frac{\begin{pmatrix} FV-min\begin{pmatrix} Q1, \\ Q2 \end{pmatrix} \\ -min\begin{pmatrix} Q3, \\ Q4 \end{pmatrix} \end{pmatrix}}{FV} \\ &\geq H(X..Y) \end{aligned} \tag{18}$$

The second shows the second estimate has strength value at least candidate coherent rules with supersets of item.

$$\begin{aligned} H'_b(X..Y) &= \frac{\begin{pmatrix} FV-min\begin{pmatrix} Q1, \\ [Q2]-[Q2] \end{pmatrix} \\ -min\begin{pmatrix} Q3+[Q2], \\ Q4 \end{pmatrix} \end{pmatrix}}{FV} \\ &= \frac{(FV-min\begin{pmatrix} Q3+[Q2], \\ Q4 \end{pmatrix})}{FV} \\ &\geq \frac{FV-min\begin{pmatrix} Q3+[Q2]+\delta c, \\ Q4+\delta d \end{pmatrix}}{FV} \\ &\geq H'_b(XE..Y) \\ &\geq \frac{\begin{pmatrix} FV-min\begin{pmatrix} Q1, \\ Q2 \end{pmatrix} \\ -min\begin{pmatrix} Q3, \\ Q4 \end{pmatrix} \end{pmatrix}}{FV} \\ &\geq H(X..Y) \end{aligned} \tag{19}$$

Similarly, the third estimate has strength value at least candidate coherent rules with supersets of item.

$$\begin{aligned} H'_c(X..Y) &= \frac{\begin{pmatrix} FV-min\begin{pmatrix} Q1, \\ [Q2]-[Q2] \end{pmatrix} \\ -min\begin{pmatrix} Q3, \\ Q4+[Q2] \end{pmatrix} \end{pmatrix}}{FV} \\ &= \frac{\left(FV-min\begin{pmatrix} Q3, \\ Q4+[Q2] \end{pmatrix}\right)}{FV} \\ &\geq \frac{(FV-min\begin{pmatrix} Q3+\delta c, \\ Q4+[Q2]+\delta d \end{pmatrix})}{FV} \\ &\geq H'_c(XE..Y) \\ &\geq \frac{\begin{pmatrix} FV-min\begin{pmatrix} Q1, \\ Q2 \end{pmatrix} \\ -min\begin{pmatrix} Q3, \\ Q4 \end{pmatrix} \end{pmatrix}}{FV} \\ &\geq H(X..Y) \end{aligned} \tag{20}$$

The strongest possible strength value is the maximum of all these estimates because we do not know exactly how the support values *Q2* decreases over the candidate coherent rules with the supersets of item but we know that in the worse case, it has a minimum value of zero. Hence, we can calculate the strongest possible strength value of a group of coherent rules with supersets of item,

$$\begin{aligned} &maxPossible\_s(X..Y) = \\ &\max\left(H'_a(X..Y), H'_b(X..Y), H'_c(X..Y)\right) \end{aligned} \tag{21}$$

Extending from the anti-monotone properties of the strongest possible strength value of candidate coherent rules *maxPossible_s*(*X..Y*), we proposed an extended approach to estimate the strongest coherent rules value. This approach complements the first approach and uses more estimates on the support values. This is explained in the next section.

3) Estimating The Strongest Strength Value of A Group of Coherent Rules – Part 2

Previously, there are three considerations to zero the estimated support value of *Q2F*. In each consideration, it still requires scans for another two support values as shown in equations 20, 21, and 22. In the current approach, we estimate all the support values *Q1*, *Q2*, *Q3*, *Q4*. And, the estimated strongest possible strength value of coherent rules is calculated based on estimated support values, *Q1F*, *Q2F*, *Q3F*, and *Q4F* without scanning for each real support value. The estimated support value *Q1F* is the total number of occurrence of an antecedent item set in the overall transaction records. The estimated support value *Q2F* is zero. The estimated support value *Q3F* is calculated as *C1-Q1F*, and *Q4F=m-C1*. This saves time to scan for real support values, and we can estimate the strongest possible strength value of coherent rules, which also inhibits anti-monotone. It follows, if the strength value is lower than the strength value needed, we can avoid generate candidate coherent rules with its supersets of item before scanning for support values. Since this estimate consists of only the estimated support values, this estimated strongest possible strength value is lower than the calculated strongest possible strength value using some of the actual support values in equation 23. Hence, this estimate is suitable to be used as the first check to stop generating candidate coherent rules before using the calculated one in equation 23.

We show the procedures how to estimate the support values.

Suppose, *Q2F*=0 and its current support value is transferred to support value *Q1F=Q1+Q2*. The estimated support value *SQF3(X..Y)=C1-Q1F*. And, the remaining support values out of total transaction records, *m* gives *SQF4(X..Y)=m-SQF1(X..Y)-SQF3(X..Y)*. (22)

None of the estimated support values requires a scan for individual support value *Q[Z]*. The value of *Q1F* is obtained via a scan through the transaction records for the occurrence of antecedent item set X alone. Such scan is less expensive than individual scan for *Q1*, and *Q2*. The estimate *Q3F*, however, can give negative values on its support value (whenever, *Q1F>C1*). We need to make further adjustment. Whenever *SQF1(X..Y)* has value larger than *C1*, we zero the estimate support value, *SQF3(X..Y)*=0, and transfer the differences to *Q4*. We show this below,

Let *Diff=C1-Q1F*
If (*Diff* <0)
  Q3F=0 (23)
  Q4F=m-Q1F- Diff
Else
  Q3F= Diff
  Q4F=m-Q1F-Q3F
End

Based on these estimates, we can now calculate the strongest possible strength value of coherent rules without scanning the individual support values. We show the anti-monotone of this new estimate,

$$\begin{aligned} maxEstPossible\_s(X..Y) &= \frac{\begin{pmatrix} FV-min\begin{pmatrix} QF1, \\ Q2-[Q2] \end{pmatrix} \\ -min\begin{pmatrix} QF3, \\ QF4 \end{pmatrix} \end{pmatrix}}{FV} \\ &= \frac{(FV-min\begin{pmatrix} Q3F, \\ Q4F \end{pmatrix})}{FV} \\ &= \frac{(FV-min\begin{pmatrix} C1-Q1-Q2, \\ m-C1-Q1-Q2 \end{pmatrix})}{FV} \\ &\geq \frac{(FV-min\begin{pmatrix} Q3F+\delta c, \\ Q4F+\delta d \end{pmatrix})}{FV} \\ &\geq maxEstPossible\_s(XE..Y) \\ &\geq \frac{\begin{pmatrix} FV-min(Q1,Q2) \\ -min(Q3,Q4) \end{pmatrix}}{FV} \\ &\geq H(X..Y) \end{aligned} \quad (24)$$

We use both the calculated and the estimated strongest possible strength values (*maxPossible_s* and *maxEstPossible_s*) and the percentage *w*, to avoid generating and validating some of the candidate coherent rules that does not meet the property. That is, if an item set does not have the estimated strongest strength value at least (1-*w*)× *max_s*, then all the subsequent candidate coherent rules with supersets of item will also not.

The percentage value (*w*) is provided by user, and can be preset to be arbitrarily small. During the generation and validation process, we can maintain the strongest coherent rules strength value found so far from a given transaction records. Hence, the decision to further generate candidate coherent rules depends on this momentary strength *max_s*. And, coherent rules found are kept in a buffer before the entire search ended. As this momentary strongest strength values increases, some coherent rules included in a buffer earlier (based on a lower strongest strength values) are discarded based on the new minimum strength value required, (1-*w*)× *max_s*. This process repeats itself, and finally coherent rules with strength value within the top *w* of the strongest coherent rules can be discovered.

The search for coherent rules hence does not require a user to specify the minimum strength value for coherent rules. The percentage value *w* supplied by a user should be distinguished from a minimum strength value required for coherent rules. Setting the latter too high or too low has adverse effect discussed in Section II. The exact and needed minimum value varies across given transaction records, and is typically unknown. Whereas, the former is a delta range of strength value from the arbitrary strongest strength value in given transaction records.

VI. 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* are 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 powerset of I_Y, but does not need to generate the complete powerset 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 supersets, 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.

VII. Experiments and Discussions

We have conducted a number of experiments. In this paper, we report the results of three 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. The second experiment shows that 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. The experiment for this purpose is conducted in the mushroom data set. 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 both zoo and mushroom 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 [23] 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 *mammalia* and *amphibian*. While mammalia 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.

Algorithm generateNextCR(candidateCoherentRules *R*, items I_Y, itemIndex PV_{X1}, itemIndex PV_{X2}, itemIndex PV_Y, itemIndex PV_{Max}, subItems *T*, orderedSet<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 $RA \leftarrow$ concatenate(PV_{X1}, RA)
      4.1.2 X $\leftarrow$ $\{i_L: L \in RA\}$
      4.1.3 Let R be the set of candidate coherent rules corresponding to (X, Y) such that $R=(X \Rightarrow Y, \neg X \Rightarrow \neg Y)$

//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 ($maxEstPossible\_s \geq min\_s$)
         4.1.5.1 Compute *maxPossible_s*, *Q1*, *Q2*, *Q3*, *Q4* using another scan
         4.1.5.2 If ($Q1>Q3$) And ($maxPossible\_s \geq min\_s$)
            4.1.5.2.1 $\forall r \in R$ compute H_r and store it
            4.1.5.2.2 Update min_s based on user-specified w and the strongest H_r found
            4.1.5.2.3 If ($H_r \geq min\_s$)
               4.1.5.2.3.1 $CR = CR \cup R$
               4.1.5.2.3.2 $toRemove = \{ cr: cr \in CR$ and $H_{cr} < min\_s \}$
               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 item set of some $r \in R$ 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 $r \in R$ 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 $RA \leftarrow (RA - PV_{X1})$ //remove an item from the buffer of ant. item set//
   4.2 End
   4.3 $PV_{X1} := PV_{X1} + 1$ //increase the first pointer value//
5. End

Algorithm 1: Generate Coherent Rules

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 mammalia 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 mammalia such as goat has no feather but has milk and backbone therefore feather(0), milk(1), and backbone(1) are reported associated with mammalia(1). We list all rules contains not more than four items (due to length of paper) in table 2.

Table 2: Rules describe *mammalia*

Antecedent Item Set		Conseq. Item Set
milk(1)	⇒	*mam.*(1)
feathers(0),milk(1)	⇒	*mam.*(1)
milk(1),backbone(1)	⇒	*mam.*(1)
feathers(0),milk(1),backbone(1)	⇒	*mam.*(1)
milk(1),breathes(1)	⇒	*mam.*(1)
feathers(0),milk(1),breathes(1)	⇒	*mam.*(1)
milk(1),backbone(1),breathes(1)	⇒	*mam.*(1)
milk(1),venomous(0)	⇒	*mam.*(1)
feathers(0),milk(1),venomous(0)	⇒	*mam.*(1)
milk(1),backbone(1),venomous(0)	⇒	*mam.*(1)
milk(1),breathes(1),venomous(0)	⇒	*mam.*(1)

We found these rules describe mammalia correctly. In fact, the first and the shortest rule $milk \Rightarrow mammalia$ describe a fundamental characteristic of a mammalia explicitly. From literature review, the second rule may be deemed redundant in comparison with the first rule because inclusion of an additional item set, feather(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 mammalia more comprehensively at the same strength of 1.0. That is, an animal of mammalia does not have feather but milk. If we discard feather(0), we loss this item as a descriptive.

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 mammalia, 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 notebook computer Pentium Centrino 1GHz with 1.5G of main memory and running Windows XP Home Edition. 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. *Mushroom dataset*

In our next experiment, we run our search algorithm on mushroom dataset [24] which contains 8124 transactions and 119 items. To grasp the search space, if one computation cycle time is needed to form a combination, it takes at least 3E+58 years to complete. Our search for both poisonous and edible mushrooms is completed within 17 seconds with 6 rules found. We list these rules in Table 3(a) and Table 3(b).

Table 3(a): Rules describe edible mushroom

Antecedent Item Set		Conseq. Item Set
odor.almond	⇒	*Edible*
odor.almond, stalk-color-below-ring.orange	⇒	*Edible*

Table 3(b): Rules describe poisonous mushroom

Antecedent Item Set		Conseq. Item Set
cap-color.green, odor.spicy, gill-attachment.free	⇒	*Poisonous*
cap-color.green, odor.spicy, gill-attachment.free, stalk-color-below-ring.orange	⇒	*Poisonous*
cap-color.green, gill-attachment.free, stalk-color-below-ring.cinnamon	⇒	*Poisonous*
cap-color.green, gill-attachment.free, stalk-color-below-ring.orange, stalk-color-below-ring.cinnamon	⇒	*Poisonous*

We leave the correctness of these results to domain experts since we are no expert. The strengths of these rules are around 0.77 out of 1.0, this suggests that there may exist some exceptional cases besides these strongest rules.

In comparison, a typical association rule mining technique such as *apriori* reports more than 100 thousands of rules with confidence value at 100%. Some of these rules are not interesting, and one way to filter these are to select high confidence rules with positive leverage values. Rules with positive leverage are rules that are dependent to each other. However, after filtering high confident rules with positive leverage, it still left us more than 100 thousands of rules for this dataset. Among these rules, it contains our six rules. We conclude from these observations that our approach produces rules that are concise and easier to apply.

C. *Artificial datasets*

We follow to generate a following three dense artificial datasets with an increase in complexity using the IBM synthetic data generator [25]. 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,

i) D100T10N100L50I9,
ii) D100T10N500L50I9,
iii) D100T10N1000L50I9

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

i) D100T10N100L10000I4,
ii) D100T10N500L10000I4,
iii) 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.

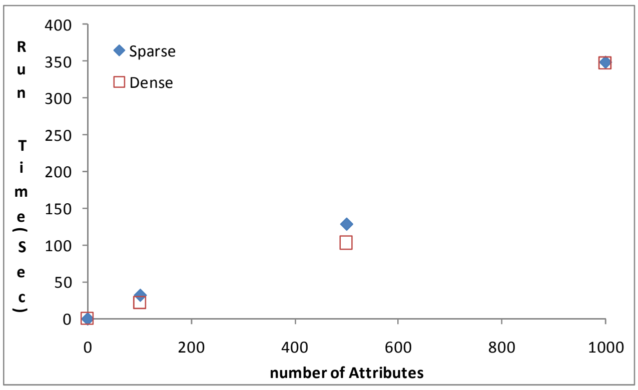

Fig 1: Search time on an increase complexity on dense and sparse dataset

VIII. 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.

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] Huynh, Xuan-Hiep, Guillet, Fabrice and Briand, Henri. Evaluating Interestingness Measures with Linear Correlation Graph. *Advances in Applied Artificial Intelligence.* Springer Berlin / Heidelberg, 2006, Vol. 4031, pp. 312-321.
[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] Koh, Yun Sing and Rountree, Nathan. Finding Sporadic Rules Using Apriori-Inverse. *Lecture Notes in Computer Science: Advances in Knowledge Discovery and Data Mining.* Springer Berlin / Heidelberg, 2005, Vol. 3518, pp. 97-106.
[7] Koh, Yun Sing, Rountree, Nathan and O'Keefe, Richard. Advances in Knowledge Discovery and Data Mining. *Lecture Notes in Computer Science.* Springer Berlin / Heidelberg, 2006, Vol. 3918, Mining Interesting Imperfectly Sporadic Rules, pp. 473-482.
[8] Castelo, Robert, Feelders, Ad and Siebes, Arno. Advances in Intelligent Data Analysis: 4th International Conference Proceedings. *Lecture Notes in Computer Science.* Springer Berlin / Heidelberg, 2001, Vol. 2189, MAMBO: Discovering Association Rules Based on Conditional Independencies, pp. 289-298.
[9] Lin, Wen-Yang, Tseng, Ming-Cheng and Su, Ja-Hwung. *A Confidence-Lift Support Specification for Interesting Associations Mining.* Springer-Verlag, 2002. pp. 148-158.
[10] Brin, Sergey, Motwani, Rajeev and Silverstein, Craig. *Beyond market baskets: generalizing association rules to correlations.* ACM, 1997. Vol. 26, pp. 265-276.
[11] 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.
[12] 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.
[13] Piatetsky-Shapiro, G. *Discovery,analysis,and presentation of strong rules.* AAAI/MIT Press,Menlo Park,CA, 1991. pp. 229-248.
[14] 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.
[15] Webb, G. I. and Zhang, S. *k-Optimal Rule Discovery.* Springer Netherlands, 2005, Data Mining and Knowledge Discovery, Vol. 10(1), pp. 39-79.
[16] Rescher, Nicholas. *Conditionals.* Cambridge, Massachusetts : The MIT Press, 2007. pp. 1-58.

[17] Cornelis, C., et al. *Mining Positive and Negative Association Rules from Large Databases.* Bangkok, Thailand, June 2006. pp. 1-6.
[18] Silverstein, Craig, Brin, Sergey and Motwani, Rajeev. *Beyond Market Baskets: Generalizing Association Rules to Dependence Rules.* Kluwer Academic Publishers, 1998, Data Min. Knowl. Discov., Vol. 2, pp. 39–68.
[19] Goodman, L. A. & Kruskal, W. H. *Measures of Association for Cross Classifications.* Journal of the American Statistical Association, American Statistical Association, 1954, 49, 732-764.
[20] Kviz, Frederick J. *Interpretating Proportional Reduction in Error Measures as Percentage of Variation Explained.* The Sociological Quarterly, 1981. Vol. 22, pp. 413-420.
[21] Hubert M. Blalock, Jr. *Social Statistics.* New York : McGraw-Hill Book Company, 1972. pp. 302-303.
[22] Sim, Alex Tze Hiang, Indrawan, Maria and Srinivasan, Bala. *Mining Infrequent and Interesting Rules from Transaction Records.* Cambridge, UK : WSEAS, 2008. Advances on Artificial Intelligence, Knowledge Engineering, and Data Bases (AIKED 2008). pp. 515-520.
[23] AILab. Zoo dataset. [Online] January 2007. http://magix.fri.uni-lj.si/orange/doc/datasets/zoo.htm.
[24] Frequent Itemset Mining Dataset Repository. [Online] http://fimi.cs.helsinki.fi/data/.
[25] IBM Quest Data Generator. [Online] July 1997. www.almaden.ibm.com/cs/quest/syndata.html.