# A Comparative Study of Apriori and Rough Classifier for Data Mining

MOHAMAD FARHAN MOHAMAD MOHSIN,
AZURALIZA ABU BAKAR & MOHD HELMY ABD WAHAB

ABSTRACT

*This paper presents a comparative study of two data mining techniques; apriori $A_C$ and rough classifier $R_c$. Apriori is a technique for mining association rules while rough set is one of the leading data mining techniques for classification. For the classification purpose, the apriori algorithm was modified in order to play its role as a classifier. The new apriori called $A_C$ is obtained through the modification of the frequent item set generation function and a filtering function is proposed. The purpose of this modification is to consider the apriori as a target-oriented training where target class is included during mining. Frequent item set generation phase is carried out to mine all attributes together with target class. The performance of $A_C$ is compared with a rough classifier. Rough classifier $R_c$ is chosen for comparison for its rule based structure. Three important measures will be used for both techniques, the accuracy of classification, the number of rules, and the length of rules. The experimental result shows that $A_C$ is comparable with $R_C$ in terms of accuracy and in several experiments it performs better. $A_C$ produced more rules than $R_C$. This study indicates that apriori can be used as an alternative classifier.*

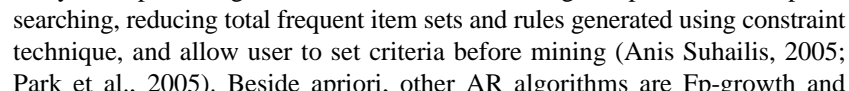*Keywords: Rough set, apriori algorithm, rule based classifier.*

ABSTRAK

*Kertas ini membentangkan kajian perbandingan di antara dua teknik perlombongan data; pengelas apriori $A_C$ dan pengelas kasar $R_c$. Apriori merupakan teknik untuk melombong petua sekutuan manakala set kasar adalah satu dari teknik pengelasan yang terkemuka. Untuk tujuan pengelasan, algoritma apriori telah diubahsuai untuk ia memainkan peranan sebagai pengelas. Apriori yang baru dipanggil $A_C$ diperoleh melalui pengubahsuaian fungsi penjanaan item kerap dan satu fungsi penapisan dicadangkan. Tujuan penbgubahsuaian ini ialah untuk mempertimbangkan apriori sebagai set latihan berorientasikan sasaran iaitu kelas sasaran digunakan semasa*

*perlombongan. Fasa penjanaan item kerap dilaksanakan untuk melombong semua atribut bersama dengan kelas sasarannya. Prestasi $A_C$ dibandingkan dengan pengelas kasar. Pengelas kasar $R_C$ dipilih untuk perbandingan kerana struktur berasaskan petuanya.Tiga pengukuran penting digunakan untuk kedua-dua teknik, ketepatan pengelasan, bilangan petua, dan panjang petua. Hasil ujikaji menunjukkan $A_C$ adalah setanding dengan $R_C$ dalam ketepatan, dan dalam beberapa ujikaji, prestasinya lebih baik. $A_C$ menghasilkan lebih banyak petua dari $R_C$. Kajian ini menunjukkan yang apriori boleh digunakan sebagai satu pengelas alternatif.*

*Katakunci: Set kasar, algoritma apriori, pengelas berasaskan petua.*

## INTRODUCTION

Association rule (AR) is a data mining technique to find interesting relationship and construct rule within frequent attributes in database (Han and Kamber, 2001). Agrawal et al. (1993) originates the research in AR by introducing apriori algorithm as first algorithm to analyze market basket problem. They had attracted researchers over the world to explore it and in recent times, the study on apriori algorithm focuses on increasing its performance speed, searching, reducing total frequent item sets and rules generated using constraint technique, and allow user to set criteria before mining (Anis Suhailis, 2005; Park et al., 2005). Beside apriori, other AR algorithms are Fp-growth and ARSC (Han et al., 2000; Brian et al., 1997).

Recent studies indicate that apriori algorithm had been widely used to perform classification. The experimental result demonstrated by Liu et al. (1998) confirmed that AR can be used as classifier. They modified the first apriori algorithm to play a role as classifier by suggesting Classification Based on Association (CBA) algorithm. As classifier, CBA applies separate-and-conquer approach by building up the interesting rules in greedy fashion. The set of interesting rules were generated based on minimum and confidence support threshold. As a sequence, CBA was modified to improve its efficiency, such as increasing the mining speed, better rules arrangement method, and lower rule quality (Liu et al., 2000; Davy et al., 2005). Beside CBA, Multi Class Association Rule (MCAR) and Apriori-C are also two associative classification algorithms that utilized apriori based concept (Thabtah et al. 2004; Jovanoski & Lavrac, 2001). Both algorithms provide better classification accuracy compared to CBA. Other associative classification algorithms are CMAR, CSFP, and GAIN are algorithms which provide better classification accuracy however those methods do not employ support and confidence criteria in apriori approach. As a result, they can reduce the mining time and increased the classification accuracy (Li et al., 2001; Yudho & Raj, 2004; Chen et al., 2005).

2

In most research, the performances of classification algorithms based on AR are compared within the associative classification algorithms classes which are based on AR technique or traditional classification algorithm. The comparative study should also be carried out with other leading classifiers that have similar structure of knowledge such as rough classifier and decision tree, C4.5 (Liu et al., 2000; Thabtah et al., 2004; Davy et al., 2005; Li et al., 2001). Theoretically, two classification techniques will generate different sets of rules via knowledge even though they are implemented to the same classification problem and perform well in classification with comparable accuracy. This is supported by Geng & Hamilton (2006) whereby different rule sets can be found when various classification algorithms are tested on the same dataset. According to Ma et al. (2000), each rule may give different knowledge and some of the rules produce consistent information. The results from different classification system can be used to determine appropriate technique for particular domain. Previously, Zorman et al. (2002) compared the result between AR and decision tree in terms of rule quantity after filtering and reduction. The experiments showed that the decision tree produced smaller rule than AR. Besides that, Mohsin and Abdul Wahab (2008) evaluated rough classifier and decision tree classification technique towards several UCI dataset. They indicated that rough classifier is able to generate consistent accuracy, shorter rule and higher coverage. However, the quantity of rule produced by decision tree is much lower than rough classifier.

According to Bakar (2002), the performance of a classification system cannot be based only on higher accuracy, but also the quality of knowledge such as minimum number of rules, rule length, and rule strength that also needs to be assessed. A good rule set must have a minimum number of rules and each rule should be as short as possible. Moreover, an ideal model should be able to produce fewer rule with shorter rule and classify new data with good accuracy. Therefore, the quality of knowledge will determine the classifier to classify new cases with good accuracy.

In this study, the performance of a modified AR classifier will be compared with rough classifier. The rough classifier is chosen as classification technique due to comparable structure of knowledge generated with the AR classifier. The purpose of this study is to investigate the accuracy and the quality of rule in terms of size of rules in both approaches. The rules will be thoroughly investigated for its responds to the positive and negative class, and the maximum and the minimum length.

This paper is organized as follows. Section 2 outlines the basic notion of AR and the associative classification. The proposed new algorithm for $A_C$ is discussed in section 3. The experiment and result will be presented in section 4 and the final section concludes this work.

In this section, the basic of association rule mining and rough classification modeling is discussed. Association rule mining or AR mining is the identification of frequent items that occur in a database of transaction. Each item ($i_j$) in a transaction is an important feature that contributed to the computation of item set and generation of rules. Basically, let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of item and $D$ be a set of transactions, where each transaction $T$ is a set of items such as that $T \subseteq I$. An AR is an implication of form $X \rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$. The rule $X \rightarrow Y$ has support s in the transaction $D$ if $s\%$ of transactions id D contain $X \cup Y$. The rule $X \rightarrow Y$ holds in the transaction with confidence $c$ if $c\%$ of transaction in $D$ that contain $X$ also contain $Y$. AR mining's processes begin with searching for frequent item set with user-specified minimum support and later rules are contrasted by binding the frequent item with its values and class. Strong rules are defined as rules that have confidence more than the minimum confidence threshold.

Suppose a training dataset $D$ of a set of instances $I = \{i_1, i_2, \ldots, i_m\}$ follows the schema $(A_1, A_2, \ldots, A_n)$, where $A$ is an attribute of type categorical or continuous. For a continuous value, we assume that its value range is discretised into intervals, and the interval is mapped to a set of consecutive positive integers. By doing so, all the attributes are treated similar to an item in a market basket database. Let $C = \{c_1, c_2, \ldots, c_m\}$ be a set of class labels. Each data object in D has a class label $c_i \text{Œ} A$ associated with it. A classifier is a function from $(A_1, A_2, \ldots, A_n)$ to C. Given a pattern $P = \{p_1, p_2, \ldots, p_m\}$, a transaction $T$ in $D$ ($T \subseteq I$) is said to match a pattern $P$ if and only if for ($1 \leq j \geq k$), the value of each $i_j$ in $T$ match with $p_m$. If a pattern $P$ is frequent in class C then the number of occurrences of $P$ in $C$ is satisfy $support\ (R)$ of the rule $R: P \rightarrow C$. The ratio of the number objects matching pattern $P$ and having class label $c$ versus the total number objects matching pattern $P$ is called the confidence of $R$, *confidence (R)*.

Rough classifier was inspired by the concepts of the rough set theory defined by Pawlak (1993) in the early 1980's is a framework for discovering relationship in imprecise data. Rough classifier was developed by Lenarcik & Piasta (1994) are similar in concepts to the feature subsets of Kohavi & Frasca (1994). The primary goal of the approach is to derive rules from data represented in an information system. The results from training set usually a set of propositional rules that may have syntactic and semantic simplicity for a human. The rough set approach consists of several steps leading towards the final goal of generating rules from the information or decision systems. The first step involves mapping of information from the original database into the decision system format. In rough classification modeling, the data are required to be discretised. Consequently the data preprocessing is one of the

4

important and tedious task to be performed. The next step will be the computation of reducts from data. Reducts are set of important attributes found within the information system that can be considered more important than other attributes. Reduct computation involve complex process and in addition there are several techniques to compute reducts such as genetic algorithm, Johnson reducer, and dynamic reducts. The knowledge in rough set is obtained through rules generation from reducts. Rules are constructed by binding together the attributes and its values in particular classes. These rules play important roles as a classifier and will be used to classify new unseen data or cases. Detail theory and concept on rough classification technique can be found in Pawlak et al. (1993).

## MODIFIED APRIORI CLASSIFER ($A_C$)

$A_C$ is based on apriori algorithm (Agrawal & Srikant, 1994). Basically, apriori contains two phases; frequent item set generation and rule generation. The new apriori classifier involves the modification in frequent item set generation and new filtering function was added during the rule generation phase. The purpose of this modification is to treat apriori as a target-oriented training where target class is included during mining.

Frequent item set generation phase is set to mine all attributes together with target class. $A_C$ consider the last attribute as target class and during mining, all candidates and frequent item sets are kept in tables where the last item of the column is a target class. Let say $D$ is a training dataset with target class $Z = \{c_1, c_2, ..., c_n\}$, collection of attributes $A = \{A_1, A_2, ..., A_n\}$ with the value of each attribute $A_n = \{av_{n1}, av_{n2}, ..., av_{nx}\}$. $A_C$ will generate a list of candidates where the last set of the list is target class value, $C_k = \{ av_{nx}, bv_{nx}, cv_{nx}, d_{nx}, ..., c_n\}$. Similar to apriori, all candidates in $C_k$ must pass the minimum support threshold before it can be regarded as a frequent item set.

To minimize mining time, all attribute's value are transformed to a unique format which sorted ascending before mining process. For example attribute sex which contain value 'male' and 'female' and attribute race which valued as 'Malay', 'Chinese', and 'Indian' are transformed to the value of *D01* (male), *D02* (female), *D03* (Malay), *D04* (Chinese), and *D05* (Indian). The formalization of unique attribute value's name format as well as the allocation of last item in candidate list $C_k$ particularly for decision class makes $A_C$ modeling different with others associative classification based on apriori. Algorithm 1 is a modified frequent item set generation where the function to scan target class before mining is inserted at line 2. The output of first phase is frequent item set.

5

```
Input: Transaction database , D, minimum_support_value
Output: Frequent item set in D , L

Method:

(1)        L₁=find_frequent_itemset(D)
(2)        Get target class value , Z
(3)        for(k=2; Lₖ₋₁ ; k++) {
(4)          Ck=apriori_gen(Lₖ₋₁, minimum_support_value)
(5)            for each transaction t∈D { //scan D for counts
(6)          Cₜ=subset(Cₖ,t); // get subsets of t that are candidates
(7)                for each candidate c∈Cₜ
(8)                    c.count++;
(9)            }
(10)         Lₖ= {c ∈ Cₖ | c.count ≥ minimum_support_value }
(11)       }
(12)       return L=∪ₖLₖ
```

ALGORITHM 1. Modification on frequent item set generation phase

Algorithm 2 indicates the rule generation phase of modified $A_C$. During this phase, a new filtering function is added in this algorithm. This function will examine the last item of frequent item set and identify the target class (line 3). If the last item is the target class, $A_C$ will eliminate the frequent item set while generating rules for those items with a target class. Consider the

```
Input: Frequent item set; L, minimum confidence value ; C
Output: Strong classification rule ; r

Method:

Generate classification rule function (L; frequent item set, C; minimum confidence ){

(1)        R=Ø;
(2)        for each frequent item  I in L {
(3)            for each non-empty subset A in I and with target class {
(4)                    generate classification rule
(5)                count confidence (A⊗I-A) =support(I)/support(A)
(6)                if confidence >confidence_minimum
(7)                      scan redundant rule {
(8)                            if redundant rule
(9)                            select rule with confidence > highest &
                          accept as a strong rules
(10)                           if none
(11)                                accept as a strong rules
(12)                      }
(13)             rules r="a⊗ I-a";
(14)             R=R_{r};
(15)         }
(16)}
```

ALGORITHM 2. Rule generation phase of $A_C$

6

frequent item set, $L_k=\{av_{n1},\ av_{n2},...\ av_{nx}\}$. If the last item of $L_k$ œ Z, then it will be removed from the system. If pass, rule will be generated if and only if the confidence of the rule is higher than threshold. Beside that, filtering function will also identify the redundant rules. Once the function detects redundant rules, $A_C$ will select one rule with the highest confidence value as strongest rule (line 7-12).

Figure 1 depicts the phases in $A_C$ rule generation process. Process no 1-10 is a frequent item set generation phase and process no 11-18 is a rule generation phase.
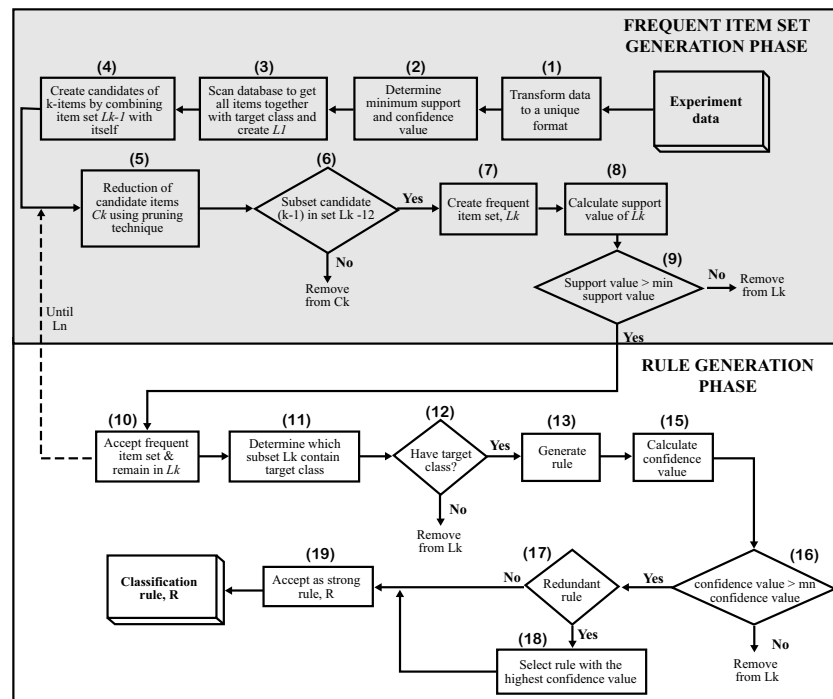


FIGURE 1. $A_C$ rule generation process

EXPERIMENTS

Five data sets from University of California Irvine (UCI) (Murphy 1997); Breast Cancer (BRS), Australian Credit Card (AUS), Diabetes (DBS), Hepatitis (HPT), and Heart Diseases (HRT) were used in this study. For both $A_C$ and $R_C$ the data preparation and classification modeling uses the same approach. During preprocessing, the entire datasets were pre-processed where all unknown numeric attributes were replaced with mean value while max value

for character attributes. The data were discretised using boolean reasoning technique (Nguyen, 1998). The data were split into training and testing using *n*-fold cross-validation technique where 9 folds of data is prepared based on ratio of training and testing; 90:10, 80:20, 70:30, 60:40, 50:50, 10:90, 20:80, 30:70, and 40:60. $A_C$ was developed using Microsoft Visual Basic and $R_C$, used Rosetta v3.2, the Rough Set Data Analysis (Ohrm, 2002).
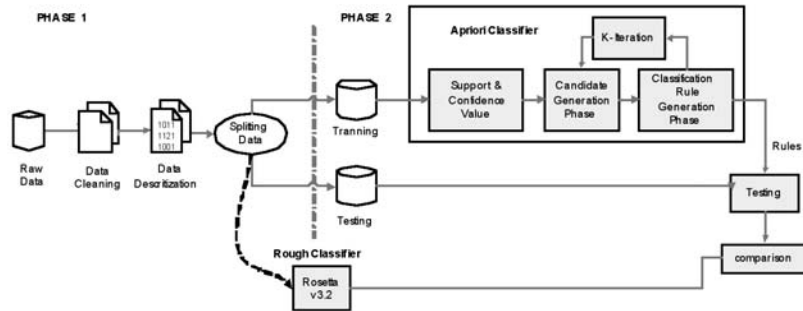


FIGURE 2. Modeling process of $A_C$ and $R_C$

$A_C$'s minimum support was set to 60% and 5% for minimum confidence during mining. The values were the best parameter which determined during primarily experiment conducted on the tested dataset. The determination of a minimum support and confidence value is the most important criteria in AR since it will control rule generation (Liu at al., 2000). If minimum support is set too high, there is a possibility for frequent item set that does not satisfy minimum support but with low confidence, large amount of frequent items will be generated (Liu at al., 1998). Both classifiers used the same training and testing set of data.

In this section, the experimental result of $A_C$ and $R_C$ are discussed. $A_C$ was compared with $R_C$ in classification accuracy, number of rules generated, and distribution of rules. The goals and notation of the experiments is formalized. The apriori classifier model and rough classifier model are given as $A_C$ and $R_C$ respectively. The training and testing data are given as *tr* and *ts*. The accuracy and the number of rule generated for both classifiers are given as *acc* and *nr*. The rule that classified to positive and negative case are given as +*re* and –*re*. Table 1 depicts the experimental result from the best model obtained from $A_C$ and $R_C$. The selection of the best model is based on three criteria; higher accuracy, fewer rules generated, and larger total testing data involved. The capital b_*NR* and b_*ACC* indicate the best model from particular classifier ($A_C$ and $R_C$) in terms of accuracy and number of rules respectively.

8

TABLE 1. Experimental Results for $A_C$ and $R_C$.

| | $m$ tr:ts | $A_C$ acc | nr +re | nr -re | $m$ tr:ts | $R_C$ acc | nr +re | nr -re | b_NR | b_ACC |
|---|---|---|---|---|---|---|---|---|---|---|
| AUS | 30:70 | 81.3 | 6446 | | 10:90 | 81.2 | 1254 | | $R_C$ | $A_C$ |
| | | | 4323 | 2123 | | | 584 | 670 | | |
| BRS | 20:80 | 91.6 | 994 | | 10:90 | 89.3 | 207 | | $R_C$ | $A_C$ |
| | | | 969 | 25 | | | 115 | 92 | | |
| DBS | 10:90 | 67.3 | 490 | | 10:90 | 71.6 | 548 | | $A_C$ | $R_C$ |
| | | | 441 | 49 | | | 296 | 252 | | |
| HRT | 30:70 | 80.9 | 7493 | | 10:90 | 73.5 | 327 | | $R_C$ | $A_C$ |
| | | | 4436 | 3057 | | | 169 | 158 | | |
| HPT | 20:80 | 79.5 | 9631 | | 20:80 | 82.3 | 248 | | $R_C$ | $R_C$ |
| | | | 0 | 9361 | | | 149 | 99 | | |

The two rightmost columns in Table 1 indicate the best in two categories; the best model based on the number of rules (NR) and the best model on accuracy (ACC). $R_C$ is more likely to be the best model of quantity rule generation because it had generated fewer rules than $A_C$ for four datasets with significant difference. $A_C$ produced more rules in using a small training data during training. Meanwhile, $R_C$ generates fewer rules from mining a larger amount of data. $A_C$ gives higher accuracy than $R_C$ in three of the cases depicted by ACC but these differences are not statistically significant. $R_C$ seems to perform better with simple and fewer rules, indicating that good decision can still be made within limited knowledge and not necessary more knowledge contributed to the decision. The rules via knowledge generated by $A_C$ may have redundancy. Therefore, $R_C$ can be considered as good classifier as well as $A_C$. $R_C$ can successfully classify more testing data (10:90) and gives the same result as $A_C$. A mathematical formula is defined to select the best model among $A_C$ and $R_C$ and is written as follows:

$$\forall \ R_C, \ A_C R_C \rangle_{better} \ Ac \rightarrow acc \ (R_C < A_C)$$
$$\forall \ R_C, \ A_C R_C \rangle_{better} \ Ac \rightarrow nr \ (R_C > A_C)$$

$R_C$, is selected as better if the NR of $R_C$ is fewer than $A_C$ and the ACC of $R_C$, is higher than $A_C$. Similarly the formula applies to $A_C$. From the experiments, the mathematical formulas define the best model among $R_C$ and $A_C$ is written as follows:

9

$$acc\ (R_C \rangle A_C) \quad \Leftrightarrow b\_ACC(R_C)$$
$$nr\ (R_C \langle A_C) \quad \Leftrightarrow b\_NR(R_C)$$

We extend our analysis by investigating the response of the rules in both techniques towards positive response (+re) and negative response (-re). Positive and negative response refers to the precedence (THEN) part of rule which act as target class. The aim of this analysis is to inspect how both classifiers are able to generate balance rule for different if the problem contain unbalance target class. From the experiment, it shows that $R_C$ generates equal number of rules for both +re and –re case but $A_C$ was biased to certain case even though the distribution of target class is badly distributed. $R_C$ generates good distribution and fewer rules with high accuracy due to the default rule generation framework. Rules generated from this framework are default rules which are more compact and shorter despite the training data size involved is small (Mollestad, 1997). For $A_C$, rule production are dependent on three factors; the number of interesting attribute combination contain in dataset, the distribution of target class in dataset whether it is equal or bias for certain class, and the setting of minimum support and minimum confidence. $A_C$ has the possibility to generate more rules although from a small training data but contain many interesting attribute combinations. Both classifiers have their strength to be selected for mining purposes.

## CONCLUSIONS

In this paper, a modified apriori classifier $A_C$ is proposed. Then a comparative study is carried out for the $A_C$ classifier and the rough classifier $R_C$ in terms of *ACC* and *NR*, and rules distribution for positive and negative cases. The experimental result shows that $A_C$ is comparable to $R_C$ in several measures. In most dataset $A_C$ is competitive to $R_C$ in classification where several models indicate that $A_C$'s accuracy is higher than $R_C$. However, in most cases $A_C$ generates more rules than $R_C$ besides DBS dataset. The significant variant in rule generation is due to the way they handle the data. $A_C$ views frequency of interesting attribute as an important issue while $R_C$ considers the distinction between attribute's values during reduction interesting through the discernibility concept. In conclusion, the modified apriori has potential to be used as an alternative classifier.

### REFERENCES

Agrawal, R. & Srikant, R. 1994. Fast algorithm for mining association rules. *Proc. Int. Conf. Very Large Databases*, pp. 487-449.

Agrawal, R., Imeilinski, T. & Swami, A. 1993. Mining association rule between sets of item in large databases. *Proc. ACM SIGMOD Conf. on Mngt. of Data*, pp. 207-216.

Anis Suhailis Abdul Kadir. 2005. Kaedah kueri dua fasa dalam algoritma apriori untuk perlombongan petua sekutuan. Master Thesis. Bangi: Universiti Kebangsaan Malaysia.

Bakar, A.A. 2002. Propositional satisfiability method in rough set classification modeling for data mining, Ph. D Thesis. Serdang: Universiti Putra Malaysia.

Brian, L., Swami, A. & Jennifer, W. 1997. Clustering association rules. *Proceedings of the 13th IEEE International Conference on Data Engineering,* Birmingham,pp 220-231

Chen, G., Liu, H., Yu, L., Wei, Q. & Zhang, X. 2005. A new approach to classification based on association rule mining. *Decision Support System* 111222.

Davy, J., Wets, G., Tom, B. & Koen, V. 2005. Adapting the CBA algorithm by means of intensity of implication. *Informatics Sciences* 175: 305-318.

Geng, L & Hamilton, H.J. 2006. Interestingness measures for data mining: A survey, *ACM Computing Survey* 38(3), pp. 1-32.

Han, J. & Kamber, M. 2001. *Data mining concept and technique*. San Francisco: Morgan Kaufmann Publishers.

Han, J., Pei, Y. & Yin, Y. 2000. Mining frequent patterns without candidate generation. *Proc. of the 2000 ACM SIGMOD Int. Conf. on Management of Data,* pp. 1-12

Jovanoski, V. & Lavrac, N. 2001. Classification Rule Learning with APRIORI-C. *In Progress in Artificial Intelligence, Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, pp. 111-135.

Kohavi, R. & Frasca, B. 1994. Useful feature subsets and roughs set reducts. 3rd International Workshop on Rough Sets and Soft Goungating (RSSC'94). San Jose. USA, ed. T.Y. Lim and A.M. Wildberger. 1-8.

Lenarcik, A. & Piasta, Z. 1994. Rough classifiers. *In Rough Set, Fuzzy Set, and Knowledge Discovery, ed. W. Zairko*. Springer-Verlag, London, pp. 298-316

Li, W., Han, J. and Pei, J. 2001. CMAR: Accurate and efficient classification based on multiple class association rules. (on-line). dfzSzcmar01.pdf/li01cmar.pdf" http://citeseer.ist.psu.edu/cache/papers/cs/27045/http:zSzzSzwww-faculty.cs.uiuc.eduzSz~hanjzSzpdfzSzcmar01.pdf/li01cmar.pdf (25 Mac 2006).

Liu, B., Hsu, W. & Ma, Y. 1998. Integrating classification and association rule mining. *Proc. Int. Conf. on Knowledge Discovery and Data Mining*, pp. 487-489.

Liu, B., Ma, Y. & Wong, C.K. 2000. Improving an association rule based classifier. *LNAI* 1910: 504-509.

Ma, Y., Liu, B., Wong, C.K., Philip, S.Y. & Lee, S.M. 2000. Targeting right student using data mining, *ACM KDD, 2000*, pp. 457-464.

Mohsin, M.F. & Abd Wahab, M.H. 2008. Comparing knowledge quality in rough classifier and decision tree classifier. *Proceeding of 3rd IEEE International Symposium of Information Technology (ITSIM08),* Kuala Lumpur, August 26-29, 2008, pp. 1109-1114

Mollestad, T. 1997. A rough set approach to data mining: Extracting logic of default rules from data. Ph. D. Norwegian University of Science and Technology.

Murphy, P.M. 1997. UCI repositories of machine learning and domain theories. (on-line). http://www.ics.uci.edu/~mlearn/MLRepository.html (10 Nov 2005).

Nguyen, H.S. 1998. Descretization problem for rough set methods. *Proc of First Int. Conf. on Rough Set and Current Trend in Computing*, pp. 545-552.

11

Ohrm, A. 2000. ROSETTA technical reference manual. (on-line). http://rosetta.lcb.uu.se/general/resources/manual.pdf (23 Feb 2006).

Park, J.S., Chen, M. & Yu, P.S. 1995. An effective hashed based algorithm for mining association rule. *ACM SIGMOD Intl. Conf. Management of Data*, pp. 201-231.

Pawlak, Z. 1993. Rough set and data analysis. *IEEE*. 1-6.

Pawlak, Z., Grzymala, B.J., Slowinski, R. & Ziarko, W. 1995. Rough Set. *Communications Of The ACM* 38(11): 89-85.

Thabtah, F., Cowling, P. & Peng, Y. 2004. MCAR: Multi-class Classification based on Association Rule Approach. *Proc of the 3rd IEEE Int. Conf. on Computer Systems and Applications*, pp. 1-7.

Yudho, G.S. & Raj, P.G. 2004. Building more accurate classifier based on strong frequent patterns. *LNAI* 3339: 1036-1042.

Zorman, M. Masuda, G. Kokol, P. Yamamoto, R. & Stiglic, B. 2002. Mining diabetes database with decision trees and association rules. *Proceedings of the 15th IEEE Symposium on Computer-Based Medical Systems, (CBMS 2002)*. pp: 134 - 139

Ziarko, W. 1999. Discovery through rough set theory. *Communications of the ACM*. 42(11): 54-57.

Mohamad Farhan Mohamad Mohsin
College of Arts & Sciences,Universiti Utara Malaysia,
06010 UUM Sintok, Kedah, Malaysia
farhan@uum.edu.my

Azuraliza Abu Bakar
 Faculty of Science and Information Technology,
Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Malaysia.
aab@ftsm.ukm.my

Mohd Helmy Abd Wahab
Faculty of Electrical and Electronic Engineering,
Universiti Tun Hussein Onn Malaysia
helmy@uthm.edu.my

12