Pertanika J. Sci. & Technol. 21 (1): 205 - 216 (2013)

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Modified Multi-Class Classification using Association Rule Mining

Yuhanis Yusof* and Mohammed Hayel Refai

School of Computing, UUM College of Arts and Sciences, Universiti Utara Malaysia, 06010 Sintok, Kedah, Malaysia

ABSTRACT

As the amount of document increases, automation of classification that aids the analysis and management of documents receive focal attention. Classification, based on association rules that are generated from a collection of documents, is a recent data mining approach that integrates association rule mining and classification. The existing approaches produces either high accuracy with large number of rules or a small number of association rules that generate low accuracy. This work presents an association rule mining that employs a new item production algorithm that generates a small number of rules and produces an acceptable accuracy rate. The proposed method is evaluated on UCI datasets and measured based on prediction accuracy and the number of generated association rules. Comparison is later made against an existing classifier, Multi-class Classification based on Association Rule (MCAR). From the undertaken experiments, it is learned that the proposed method produces similar accuracy rate as MCAR but yet uses lesser number of rules.

Keywords: Classification, association rule, rule mining, rule production, data mining

INTRODUCTION

The availability of high performance computers, data collection tools and huge memory capacities has made gathering and saving huge quantities of data possible. For example, the number of sales transactions executed during one year in a large retail supermarket such as Carrefour is numerous and the amount of data on the World Wide Web (WWW) is extremely massive as well. This vast growth of stored databases provides an opportunity for new automated intelligent data analysis methods that summarizes information from these databases. The process of generating this useful knowledge is accomplished using data mining techniques. Liu *et al*. (1998) define data mining as one of the primary steps in Knowledge Discovery from Databases (KDD), which finds and generates useful

Article history:
Received: 31 March 2012
Accepted: 31 August 2012

E-mail addresses:
yuhanis@uum.edu.my (Yuhanis Yusof),
mohammdrefai@yahoo.com (Mohammed Hayel Refai)
*Corresponding Author

ISSN: 0128-7680 © 2013 Universiti Putra Malaysia Press.

hidden knowledge from large databases.

Classification using Association (CuA), which is also known as associative classification, is a research field in data mining that integrates association rule discovery and classification. CuA utilises association rule to discover knowledge and to select a subset of which to build the classifier (Thabtah *et al*., 2010a). The main goal for CuA is to construct a classifier based on the identified knowledge from labelled input. This model is later used to predict the class attribute for a test data case (Baralis *et al*., 2008).

In the last few years, several CuA algorithms have been developed such as CPAR (Yin & Han, 2003), Live and Let Live (L^3G) (Baralis *et al*., 2004), MCAR (Thabtah *et al*., 2005), CACA (Tang & Liao, 1998), BCAR (Yoon & Lee, 2008) and others. These research studies have shown that the approach produces better classifiers (in terms of accuracy) compared to traditional classification data mining approaches like probabilistic (Meretakis & Wuthrich, 1999) and decision tree (Quinlan, 1998). Nevertheless, the CuA algorithm suffers from exponential growth of rules, i.e. they derive large numbers of rules which make the resulting classifiers being outsized and consequently limit their uses since decision makers face difficulty in understanding and maintaining a large set of rules.

In this work, an algorithm named "Modified Multi-class Classification using Association Rule Mining" (MMCAR) is proposed to reduce the number of rules produced by association rule based algorithms. The proposed MMCAR is developed based on the existing approaches of CuA. Nevertheless, MMCAR employs a new rule production algorithm which results in only relevant rules being used to predict test cases. The training method of MMCAR scans training data sets only once (Thabtah *et al*., 2005) and the class assignment (prediction) method makes a group of rule prediction instead of utilising only a single rule.

Different datasets from the UCI repository (Merz & Murphy, 1996) have been utilised to evaluate the proposed learning algorithm and to compare it with other traditional data mining classification techniques, including decision tree (C4.5) (Quinlan, 1993), greedy classification (RIPPER) (Cohen, 1995), and MCAR (Thabtah, 2005). The measures used in the experiments for comparison are the prediction accuracy and the number of rules derived.

The paper is structured as follows: The problem of CuA and its related works are presented in CLASSIFICATION USING ASSOCIATION. The proposed algorithm is discussed in MATERIAL AND METHODS. Datasets and the experiments of using different classification algorithms are demonstrated in RESULTS AND DISCUSSION, while conclusions and suggestions for further research work are given in CONCLUSIONS.

CLASSIFICATION USING ASSOCIATION

CuA is a case of association rule discovery, in which the rule on the right hand side (consequent) is the class label, and the rule on the left hand side (antecedent) is attribute values. Thus, for the rule *R*: $X \Rightarrow Y$, (X) is a conjunction of attribute values, and (Y) is the class attribute. The ultimate goal of CuA is to extract a complete set of rules which is normally called class association rules (CARs) from the training dataset.

The Problem

In the definition of the CuA problem, we employ by Thabtah (2005). Let *T* be the input training dataset with *k* different attributes $A_1, A_2, \ldots, A_k$ and *L* is a set of class labels. A specific attribute value for A_i is represented by a_i, and the class labels of *L* are represented l_j.

Definition 1: An AttributeValue (A_i, a_i) is combination of between 1 and *k* different attributes values, e.g. < (A_1, a_1)>, < (A_1, a_1), (A_2, a_2)>, (A_1, a_1), (A_2, a_2), (A_3, a_3)>, …, etc.

Definition 2: A class association rule (CAR) is given in the following format: $(A_{i1}, a_{i1}) \wedge (A_{i2}, a_{i2}) \wedge \ldots \wedge (A_{1k}, a_i k) \rightarrow l_i$, where the antecedent is a conjunction of AttributeValues and the consequent is a class.

Definition 3: The frequency (*freq*) of a CAR in *T* is the number of cases in *T* that matches *r's* antecedent.

Definition 4: The support count (*suppcount*) of a CAR is the number of cases in *T* that matches *r's* antecedent and belongs to a class l_i for *r*.

Definition 5: A CAR (*r*) passes the *minsupp* if for *r*, *suppcount*(*r*)/ |*T*| ≥ *minsupp*, where |*T*| is the number of cases in *T*.

Definition 6: A CAR (*r*) passes *minconf* if *suppcount(r) /freq*(*r*) ≥ *minconf*.

CuA Main Steps

Fig.1 depicts the main steps used in CuA. The first step involves the discovery of frequent item set. This requires methods that find complete set of the frequent items by separating those that are potentially frequent and determine their frequencies in the training dataset (step 1). A rule will be produced if an item set exceeds the *Minconf* threshold value. The rule will be in the form of $X \rightarrow l$, where *l* is the largest frequency class associated with *X* in the training dataset (Step 2). In step 3, a selection of an effective subset of rules ordering is performed using various procedures, while the quality of the selected subset is measured on an independent (test) data set in step 4.

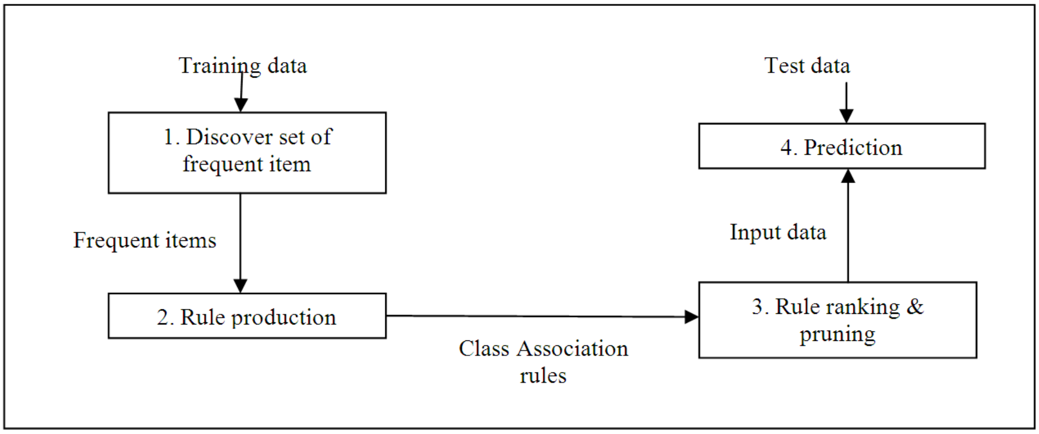


Fig.1: Main Steps in CuA (adopted from Thabtah, 2005)

To explain the CuA discovery of rules and classifier development, consider data shown in Table 1. The table consists of two attributes, A_1 (a_1, b_1, c_1) and A_2 (a_2, b_2, c_2), and two class labels (l_1, l_2). Assume that *Minsupp* = 30% and *Minconf* = 80%, the frequent one, and two items for data depicted in Table 1 are shown in Table 2, along with the associated support and confidence values. The frequent items (in bold) in Table 2 denote those that pass the confidence and support thresholds, which are later converted into rules. Finally, the classifier is constructed using a subset of these rules.

The support threshold is the key to success in CuA. However, for certain applications, rules with large confidence value are ignored because they do not have enough support. Traditional CuA algorithms such as CBA (Liu *et al*., 1998) and MCAR (Thabtah & Cowling, 2007) use one support threshold to control the number of rules derived and may be unable to capture high confidence rules that have low support. In order to explore a large search space and to capture as many high confidence rules as possible, such algorithms commonly tend to set a very low support threshold, which may lead to problems such as generating low statistically support rules and a large number of candidate rules which require high computational time and storage.

In response to these issues, one approach that suspends the support and uses only the confidence threshold to rule generation has been proposed by Wang *et al*. (2001). This confidence-based approach aims to extract all the rules in a dataset that passes the *Minconf* threshold. Other approaches are to extend some of the existing algorithms such as CBA and CMAR. These extensions resulted in a new approach called multiple supports (Baralis *et al*., 2008) that considers class distribution frequencies in a dataset and also assigns a different support threshold to each class. This assignment is done by distributing the global support

TABLE 1: Training Dataset

RowNo	Attribute_1	Attribute_2	class
1	a_1	a_2	l_1
2	a_1	a_2	l_2
3	a_1	b_2	l_1
4	a_1	b_2	l_2
5	b_1	b_2	l_2
6	b_1	a_2	l_1
7	a_1	b_2	l_1
8	a_1	a_2	l_1
9	c_1	c_2	l_2
10	a_1	a_2	l_1

TABLE 2: Potential Classifier for Data in Table 1

Frequent Items			
Rule Condition	Rule class	Supp	Conf
<a_2>	**l_1**	**4/10**	**4/5**
<a_1>	l_1	5/10	5/7

threshold to each class corresponding to its number of frequencies in the training dataset, and thus, considers the generation of rules for the class labels with low frequencies in the training data.

An approach called MCAR (Thabtah, 2005) that employs vertical mining was proposed, in which during the first training data scan, the frequent items of size one are determined and their appearances in the training data (rowIds) are stored inside an array in a vertical format. Any item that fails to pass the support threshold is discarded. The rowIds hold useful information that can be used laterally during the training step in order to compute the support threshold by intersecting the rowIds of any disjoint items of the same size. It should be noted that the proposed algorithm of MMCAR also utilises vertical mining in discovering and generating the rules. Next section describes the proposed algorithm based on an example.

MATERIAL AND METHODS

MMCAR goes through three main phases: Training, Construction of classifier, and Forecasting of new cases as shown in Fig.1. During the first phase, it scans the input data set to find frequent items in the form <AttributeValue, class> of size 1. These items are called one-item. The algorithm repeatedly joins them to produce frequent two-items, and so forth. It should be noted that any item that appears in the input dataset less than the *MinSupp* threshold will be discarded. Once all the frequent items of all sizes are discovered, the algorithm will check for the items confidence values. Only if the confidence value is larger than the *MinConf* threshold, it will then become a CAR. Otherwise, the item gets deleted. The next step is to sort the rules according to certain measures and to choose a subset of the complete set of CARs to form the classifier. Details on the proposed MMCAR are given in the next subsections.

CARs Discovery and Production

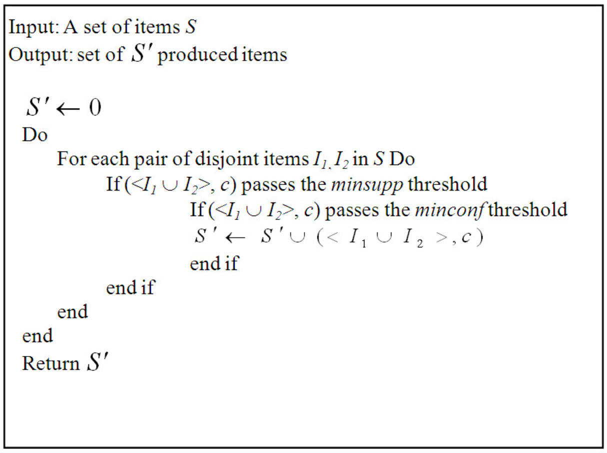
MMCAR uses an intersection method based on the so-called Tid-list to compute the support and confidence values of the item values. The Tid-list of an item represents the number of rows in the training dataset in which an item has occurred. Thus, by intersecting the Tid-lists of two disjoint items, the resulting set denotes the number of rows, in which the new resulting item has appeared in the training dataset, and the cardinality of the resulting set represents the new item support value.

The proposed algorithm goes over the training dataset only once to count the frequencies of one-items, from which it discovers those that pass the *MinSupp*. During the scan, the frequent one-items are determined, and their appearances in the input data (Tid-lists) are stored inside a data structure in a vertical format. Meanwhile, any item that does not pass the *MinSupp* will be removed. Then, the Tid-lists of the frequent one-item are used to produce the candidate two-item by simply intersecting the Tid-lists of any two disjoint one-items. Consider for instance, the frequent attribute values (size 1) ($<a_1>$, l_1) and ($<a_2>$, l_1) that are shown in Table 2 can be utilised to produce the frequent item (size 2) ($<a_1, a_2>$, l_1) by intersecting their Tid-lists, i.e. (1,3,7,8,10) and (1,6,8,10) within the training dataset (Table 1). The result of the above intersection is the set (1,8,10) in which its cardinality equals 3, and denotes the support value of the new attribute value ($<a_1, a_2>$, l_1). Now, since this attribute value support is larger than or equivalent to the *MinSupp* threshold (i.e. 30%), this 2-item will become frequent.

The above discussion describes the training approach which is called vertical mining that has been successfully used in association rule discovery (Zaki & Gouda, 2003) and in classification (Thabtah, 2005). This approach transforms the training dataset into items table that contains the locations (Tid-lists) of each item in the training dataset, employs simple intersections among these locations to discover frequent values and produces the rules. Once all the items of all sizes have been discovered, MMCAR will then check for their values and generate those which pass both the *MinSupp* and *MinConf* thresholds as items to be used in the classification association rule(s). The proposed item production algorithm is shown in Fig.2, while Fig.3 illustrates the algorithm utilized by MCAR (Thabtah, 2005).

Building the Classifier

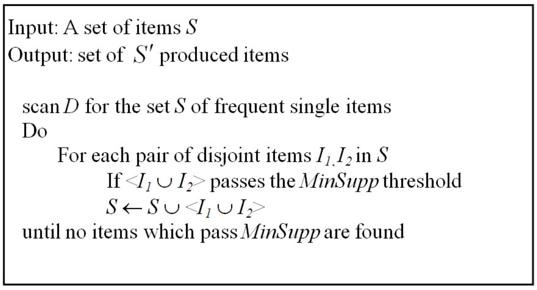
One primary limitation of CuA approach in data mining is the exponential growth of rule (Li *et al*., 2006; Thabtah *et al*., 2006). Thus, a primary motivation of this work is to cut down the number of generated CARs. Prior to pruning redundant rules and to build the classifier, rules



```
Input: A set of items $S$
Output: set of $S'$ produced items

  $S' \leftarrow 0$
  Do
       For each pair of disjoint items $I_{1,} I_2$ in $S$ Do
            If ($<I_1 \cup I_2>$, $c$) passes the minsupp threshold
                 If ($<I_1 \cup I_2>$, $c$) passes the minconf threshold
                 $S' \leftarrow S' \cup (<I_1 \cup I_2>, c)$
                 end if
            end if
       end
  end
  Return $S'$
```

Fig.2: MMCAR Item Production Algorithm



```
Input: A set of items $S$
Output: set of $S'$ produced items

  scan $D$ for the set $S$ of frequent single items
  Do
       For each pair of disjoint items $I_{1,} I_2$ in $S$
            If $<I_1 \cup I_2>$ passes the MinSupp threshold
            $S \leftarrow S \cup <I_1 \cup I_2>$
  until no items which pass MinSupp are found
```

Fig.3: MCAR Item Production Algorithm

must be sorted in order to prioritize quality rule to be chosen as a part of the classifier. In this work, the rules are sorted according to the following guidelines:

1. The rule with higher confidence is placed in a higher rank.
2. If the confidence values of two or more rules are the same, the rule with higher support will then get a higher rank.
3. If the confidence and the support values of two or more rules are the same, the rule with lesser number of attribute values in the antecedent gets a higher rank
4. If all the above criteria are similar for two or more rules, the rule which was produced first will get a higher rank.

For each sorted rule (CAR), MMCAR applies it on the training dataset, and the rule gets inserted into the classifier if it covers at least one case regardless of the rule class that is similar to that of the training case. Now, once a rule gets inserted into the classifier, all the training cases associated with it are discarded. In situations where a rule fails to cover any training case, it will then be removed. The same process is repeated until no more case remains in the training dataset or all CARs are tested.

In predicting test data case, the prediction method of MMCAR divides all the rules which match the test case into groups; one for each class label to calculate the average confidence and support values. It assigns the test case with the class of the group with the largest average confidence. In cases where there are two or more groups with similar average, the prediction method is considered for the largest average support group. Unlike other current CuA methods like MCAR (Thabtah, 2005) and CBA (Liu, 1998) which employ only the highest confidence rule for predicting the test case, our algorithm makes the prediction decision based on multiple rules (Li *et al*., 2006; Thabtah *et al*., 2006). Finally, in cases when no rules in the classifier are applicable to the test case, the default class (the majority class in the training dataset) will be assigned to that case.

RESULTS AND DISCUSSION

In this section, a recent classification using association algorithm, which is the MCAR (Thabtah, 2005), is compared against MMCAR with reference to classification accuracy and the number of rules produced by the classifier. A total of fourteen (14) UCI datasets (Merz & Murphy, 1996) have been utilized in the experiments. Cross validation which divides training dataset into (*n*+1) folds arbitrary was employed and learning was performed on the *n* folds of each iteration. Later, evaluation is undertaken on the remaining holdout fold. The process is repeated *n*+1 times and the results are obtained by taking the average values. In the experiments, the number of folds in cross validation is set at 10, as employed by Thabtah *et al.* (2005). On the other hand, the main parameters of MMCAR and MCAR, namely, *MinSupp* and *MinConf,* were set to 3% and 50%, respectively, in the experiments. This is since 3% of support usually balances between the number of rules discovered and processing time. All of the experiments were performed on Pentium IV machine with 2.0 GB RAM and 2.6 GH processors.

A.

Results and Analysis

Table 3 contains the results of the classification accuracy, while data in Table 4 depicts the number of association rules produced by MCAR and MMCAR. Data in Table 4 show similar consistency in the classification accuracies of both MMCAR and MCAR. The accuracy values obtained by both MCAR and MMCAR for six data sets (Austra, Balance-scale, Labor, Lymph, Mushroom and Wine) are at the same level; they only differ at the decimal point. For example, using Austra dataset, both classifiers generate 86% accuracy. On the other hand, MMCAR outperformed the accuracy obtained by MCAR in five data sets. Such a scenario is illustrated in Fig.4, where it shows the difference of accuracy values between MMCAR and MCAR. For example, the difference is as high as 2.85% for the Glass dataset, while there is no difference for the Labour dataset.

TABLE 3: Prediction Accuracy

Data set	MCAR	MMCAR
Austra	86.14	86.26
Balance-scale	76.96	76.17
Breast	94.99	93.83
Cleve	81.84	78.77
Glass	71.35	74.2
Heart-s	81.15	80.51
Iris	92.93	94.26
Labor	83.5	83.5
Led7	71.83	73
Lymph	78.1	78.05
Mushroom	99.6	99.67
Pima	77.12	74.44
Vote	88.2	86.39
Wine	95.73	95.73

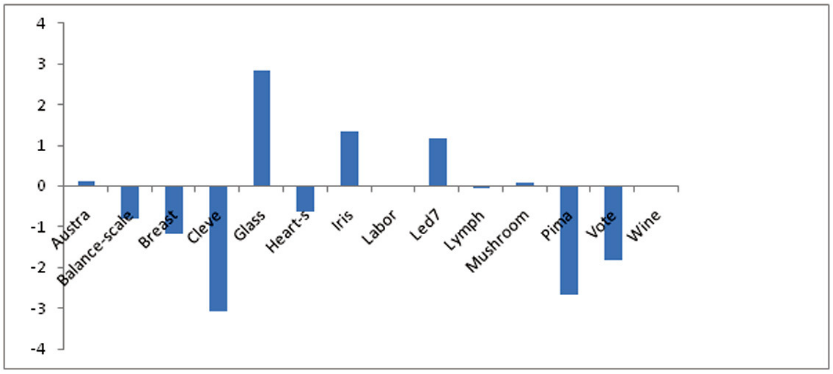


Fig.4: The Difference in Prediction Accuracy

On the other hand, the results of the generated number of rules are depicted in Table 5. Data in the table indicate that the proposed algorithm (i.e. MMCAR) produces less number of rules than MCAR. MMCAR has derived on average 63 rules for the fourteen datasets as compared to MCAR that produced an average of 76 rules. The difference of the generated rules is also illustrated in Fig.5. It is noted that MMCAR produces less number of rules for eight datasets, whereby the highest difference can be seen in the Led7 dataset.

TABLE 4: The Number of Association Rules

Dataset	MCAR	MMCAR
Austra	193	185
Balance-scale	77	81
Breast	79	80
Cleve	98	93
Glass	43	34
Heart-s	39	29
Iris	16	11
Labor	15	15
Led7	214	87
Lymph	54	54
Mushroom	42	48
Pima	107	80
Vote	83	75
Wine	11	11

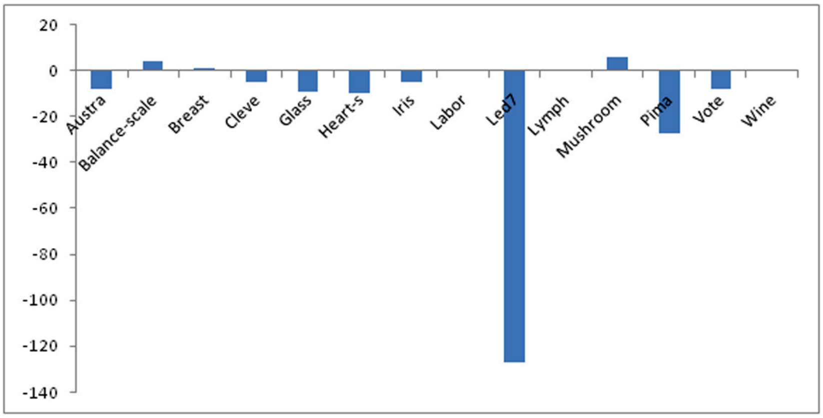


Fig.5: The Difference in the Number of Association Rules

CONCLUSION

In this paper, a modified Multi-class classification based on association rule mining is presented. The so-called MMCAR algorithm includes a new item production algorithm in generating relevant association rules. A more restricted condition is included in the item production so that MMCAR only selects the best related items in producing association rules of the classifier. Hence, a moderate size of association rules will then be obtained. Such a result will aid users in interpreting and understanding the nature of data in the context of study. The proposed MMCAR is proven to be at par with MCAR in prediction accuracy but outperforms the benchmark method in producing lesser number of association rules.

REFERENCES

Baralis, E., Chiusano, S., & Garza, P. (2004). On support thresholds in associative classification. *Proceedings of the 2004 ACM Symposium on Applied Computing*, Nicosia, Cyprus, pp. 553-558.

Baralis, E., Chiusano, S., & Garza, P. (2008). A lazy approach to associative classification. *IEEE Transactions on Knowledge and Data Engineering*, *20*, 156-171.

Cohen, W. W. (1995). Fast effective rule induction. In *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 115-123.

Li, W., Han, L., & Pei, J. (2001). CMAR: Accurate and efficient classification based on multiple class-association rules. In *Proceedings of the ICDM'01* San Jose, CA, p. 369.

Lim, T. S., Loh, W. Y., & Shih, Y. S. (2000). A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine learning, 40*, 203-228.

Liu, B., Hsu, W., & Ma, Y. (1998). Integrating classification and association rule mining. *Knowledge discovery and data mining,* 80–86.

Meretakis, D., & Wüthrich, B. (1999). Extending naïve Bayes classifiers using long item sets. In *Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, California, pp. 165-174.

Merz, C., & Murphy, P. (1996). UCI repository of machine learning databases. Retrieved from FTP of ics. uci.edu in the directory pub/machine-learning-databases.

Tang, Z., & Liao, Q. (1998). A new class based associative classification algorithm. *IAENG International Journal of Applied Mathematics, 36*(2), 136-141.

Thabtah, F., Cowling, P., & Peng, Y. (2005). MCAR: multi-class classification based on association rule. In *Proceeding of the 3rd IEEE International Conference on Computer Systems and Applications*, p. 33.

Thabtah, F., Crowling, P., & Hammoud, S. (2006). Improving rule sorting, predictive accuracy and training time in associative classification. *Expert Systems with Applications*, *31*, 414-426.

Thabtah, F. A., & Cowling, P. I. (2007). A greedy classification algorithm based on association rule. *Applied Soft Computing*, *7*, 1102-1111.

Thabtah F., Mahmood Q., McCluskey L., & Abdel-jaber H. (2010). A new Classification based on Association Algorithm. *Journal of Information and Knowledge Management*, *9*(1), 55-64.

Quinlan, J. R. (1993). *C4.5: Programs For Machine Learning*. Morgan Kaufmann.

Quinlan, J. R. (1998). Data mining tools See5 and C5.0. *Technical Report, RuleQuest Research*.

Wang, K., He, Y., & Cheung, D. W. (2001). Mining confident rules without support requirement. In *Proceedings of the Tenth International Conference on Information and Knowledge Management.* Atlanta, Georgia, pp. 89-96.

WEKA (2000). *Data Mining Software in Java*. Retrieved from http://www.cs.waikato.ac.nz/ml/weka.

Yin, X., & Han, J. (2003). *CPAR: Classification based on predictive association rules.* Paper presented at the 2003 SIAM International Conference on Data Mining, Cathedral Hill Hotel, San Francisco, CA, May 1-3, 2003. pp. 331.

Yoon, Y., & Lee, G. G. (2008). *Text Categorization Based on Boosting Association Rules*. Paper presented at the the 2008 IEEE International Conference on Semantic Computing. pp.136-143.

Zaki, M. J., & Gouda, K. (2003). *Fast vertical mining using diffsets*. Paper presented at the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington, D.C., pp. 326-335.