10<sup>th</sup> June 2016. Vol.88. No.1

© 2005 - 2016 JATIT & LLS. All rights reserved

ISSN: 1992-8645

www.jatit.org



# INFORMATION LOSS MINIMIZATION USING AUTOMATA BASED ASSOCIATIVE CLASSIFICATION

<sup>1</sup>MOHAMMAD ABRAR, <sup>2</sup>ALEX TZE HIANG SIM, <sup>3</sup>HEE JEE MEI

<sup>1,2</sup>Faculty of Computing, UTM, 81310 Skudai, Johor, Malaysia <sup>3</sup>Faculty of Education, UTM, 81310 Skudai, Johor, Malaysia

E-mail: <sup>1</sup> muhammadabrar78@gmail.com, <sup>2</sup>alex.utm@gmail.com, <sup>3</sup>hee.utm@gmail.com

# ABSTRACT

Associative Classification (AC) is one of the most prominent classification techniques that attain the attention of many researchers in last decade. AC uses the strongest association rules from the dataset in classifiers that results in improved accuracy. The exponential set of rules makes it difficult to make an efficient classifier, therefore, the rule pruning is used to reduce the number of rules during classification. This pruning result, in the loss of meaningful that might contribute to improving the accuracy. In this article, we propose the ACA based approach that minimizes the loss of information in the Associative Classification by introducing the concept of merging Automata instead of rule pruning. The merging achieves double advantages. Firstly, it reduces the set of rules as a classifier and secondly, by avoiding pruning, the risk of information loss also reduces. The experimental results over 10 datasets from the UCI repository show that the resultant classifiers are comparatively more accurate than the Jrip, LAC, AODE and others mentioned in the results.

Keywords: Associative Classification, Data Mining, Classification, Automata

# **1. INTRODUCTION**

Classification is considered as one of the main pillars in Data Mining (DM) and Machine Learning (ML) [1, 2]. It is a data analysis technique used to categorize data into different classes based on some common characteristics or associations in the data. Generally, classification consists of two basic steps i.e. a) Preparation of the classification Model - Classifier, from the available data (training dataset) and b) Classification - the prediction of the unknown class label based on the classifier model.

The DM contains a rich set of classification models; specifically, Support Vector Machine [3], Rule Based [4], Decision Tree [5, 6], Bayesian classification [1], k - Nearest Neighbour [7], and Associative Classification (AC) [8]. Among all,

the AC is relatively new and promising [9-13] as it combines the best approaches of the Association Rule Mining (ARM) and Classification. The AC is based on ARM where the first and strongest Class Association Rule (CAR) are discovered from dataset following those rules which then make the classifier model. Those stronger associations from the data in the form of CAR make the classifier more coherent and improve

# accuracy.

The AC was introduced in 1997 and immediately got the attention of many researchers in the DM area. In the last decade numerous techniques were developed by the community including e.g. CBA[14, 8], CMAR[11], CPAR [9], MCAR [13], and MAC [15]. The AC contains three basic elements i.e. a) Class Association Rule (CAR) Generation, b) Classifier Building & Rule Pruning and Rule Ranking c) Classification of unknown records using the classifier.

The CAR used by AC is a variation of the ARM whose right-hand side (Consequent) is a class label instead of an ordinary attribute. The ARM generates those CARs which pass minimum support and confidence threshold and the classifier is then built on the basis of these CARs. The AC further applies the rule ranking and pruning to minimize the number of rules in the classifier. The reason for the reduction is to make the classifier smaller in order to improve the efficiency of classification. Smaller classifier might be efficient but studies like [16, 17] showed that it reduces the accuracy. Secondly, these steps also increase the computational overhead of the classification model building phase. Furthermore, the Pruning step, if not treated wisely also eliminate rules with meaningful information

<u>10<sup>th</sup> June 2016. Vol.88. No.1</u>

© 2005 - 2016 JATIT & LLS. All rights reserved

ISSN: 1992-8645	www.jatit.org	E-ISSN: 1817-3195

[16, 17].

Beside the ample accuracy of AC, the literature also showed the following limitations.

• AC suffers from an exponential number of rules and a reasonable amount of time is paid for pruning and ranking.

• Excessive pruning always lead to a reduced accuracy by eliminating meaningful information.

• A large number of rules in classifier may increase the accuracy at the cost of efficiency and over-fitting of data.

• Almost all techniques are parameter dependent i.e. they need to set the support and confidence threshold in advance; therefore, require expert knowledge.

In the AC, most of the studies are conducted in either generating reduced numbers of CARs or improving the weighting criteria for increasing the accuracy. In this article, the focus is on reducing the size of Classifier in terms of rules of minimization with losing information [18].

In order to attain its goal, the study will focus to answer the following research question.

Q: What is the mechanism used to reduce the number of rules in Associative Classification without losing information?

To answer this question, we also need to find the solution for the following sub-questions.

1. How to merge the rules into smaller set in order to decrease the size of classifier?

2. How to handle the conflicting rules in the merged representation?

To find the solution for the above research questions, the concept of Automata from the theoretical computer sciences is used in the AC i.e. Associative Classification using Automata (ACA) introduced in [52]. In this article, we will propose a solution of how information loss can be minimized using the ACA.

# 2. LITERATURE REVIEW

The concept of classification comes from statistics which includes specifically the prediction of a categorical variable, while the other is called regression [2]. In the Data Mining and Machine Learning, classification is used in the same spirit to predict the unknown variable and its application ranging from: filling missing values [19] to the prediction and diagnosis of complex diseases [20], speech and handwriting recognition [21,22] to image analysis [23]. In this article, our focus is on the AC, Automata, and its related techniques. In order to properly cover all dimensions of the topic, this section is divided into three sub-sections. First, there is the Associative Classifier which highlights the general framework of the AC and previous work followed by the Rule Pruning and Ranking used in the AC so far, and finally the Automata and its uses in classification.

# 2.1 Associative Classification (AC)

The Associative Classification [14] is an integration of the ARM and Classification. The Associative Classifier is built on top of the Mined Association Rules that provide a stronger classifier as compared to the traditional algorithms. Due to the ample accuracy and speed of the AC, its application has spread into many areas. In the automotive industry, the AC has been used for mass customization [24]. This has resulted in best performance but low accuracy due to a small dataset.

One of the major areas of interest is malware detection where the AC achieved good results that outperformed the commercial products [25]. The Intelligent File Scoring System (IFSS), proposed by [26], is another malware detection tool which is used an ensemble approach of combining eight different classification algorithms from the family of Support Vector Machine (SVM) and AC. They generated eight different results and the final classification was completed by majority voting. The IFSS provided a very accurate result but due to heavy computation, efficiency was compromised.

Another endeavour to deal with rare classes using the AC and Hierarchical Predictive Model (HPM) was proposed for the rare classes' classification [27]. the HPM utilized a combination of weighted kNN and Na à rve Bayesian to improve accuracy. Regardless of its low efficiency, it is suitable for rare, sparse and multiclass datasets.

Another effort in the sequence is made by [28] to reduce the number of rules. This resulted in the design of a new algorithm called LC (Looking at Class). The LC joined k-item set of the "same classes" to make k+1-itemsets; otherwise, they are not joined. In the case of multiclass reference, the class with a simple majority is selected for the rule.

# 2.2 Rule Pruning And Ranking

The problem with the Association Rule Generation is by nature exponential; therefore, the AC algorithms also derive an enormous set of rules [10,29]. The number of rules is then reduced for efficient classification and eliminates those rules

10<sup>th</sup> June 2016. Vol.88. No.1

© 2005 - 2016 JATIT & LLS. All rights reserved

ISSN: 1992-8645	www.jatit.org	E-ISSN: 1817-3195

which are either redundant or not interesting, etc. There are a number of techniques called rule pruning which has been developed for rule minimization. These include specifically: redundant rules [10, 30], Database Coverage [12, 31, 32], Pessimistic Error Estimation [5, 31, 33,34], Instance-based similarity [35], Lazy Pruning [16,17,36], and conflicting rules [30, 37].

In [38], the author proposed an ACN (Associative Classifier with Negative rules) which generates both positive and negative rules for a building classifier. The rule ranking of the ACN is quite detailed and almost an integration of all techniques discussed so far. The criteria for rule ranking used by ACN are given below,

1) sup(x) > sup(y)

- 2) sup(x) = sup(y) and correlation(x) > correlation(y)
- 3) sup(x) = sup(y) and correlation(x) = correlation(y) and conf(x) > conf(y)
- 4) sup(x) = sup(y) and correlation(x) = correlation(y) and conf(x) = conf(y) and size(x) > size(y)
- 5) sup(x) = sup(y) and correlation(x) = correlation(y) and conf(x) = conf(y) and size(x) = size(y) and order(x) > order(y)

Mostly, in the literature different combinations of the above criteria are used.

# 2.3 Finite State Automata

Finite State Automata, Finite State Machine, Finite Automata or simply Automata (singular -Automaton) consist of five components and are represented formally as:  $A = \{Q, \Sigma, \delta, q0, F\}$  where A is the name of the Automata; Q represents the states of the Automata;  $\Sigma$  represents the alphabets (allowable symbols) of Automata;  $\delta$  is known as the transition function and defines rules for the transition from one state to another in the Automata based on input symbol; q0 refers to the start state and q0  $\in$  Q. Finally, F Q represents the set of final states which refers to the correct ending of the Automata.

The basic purpose of Automata is to serve as a lexical analyser compiler. This involves searching a large text body to find tokens of interest in a system that will work as a verifier, for example, the communication protocol or protocol for the secure exchange of information [51]. Generally, the Automata perform better when searching for a larger setup. This became the major reason to use the Automata as a replacement of Ranking and Pruning in the associative classification. The Automata will automatically reduce the number of rules due to its default nature of merging similar rules.

In the past, there have been successful attempts to use Automata in the Machine Learning and Data Mining. In the Natural Language Processing (NPL), the Automata were effectively applied in part of speech tagging [39]. The main reason for using Automata in NPL is its compactness and determinism. The Automata is deterministic and can be reduced to a minimal form that results in an efficient model [39].

The Automata is also used in sequence mining by [40]. They utilized the compactness of Automata to model the sequential pattern into Automata. Following this, different queries about the sequential pattern were answered by the model. Their model was also capable of producing a descriptive summary that provided help in understanding the properties of the dataset.

The Automata's default sequencing nature makes it more suitable for sequence mining. Every sequence can be considered as ordered chunks of character (string). Therefore, any such sequence pattern issue can be reduced to string matching, of which there are many efficient Automata-based algorithms in existence [41-43].

Two other variations of Automata are used in classification, specifically: Learning Automata - LA [44-46] and Fuzzy Automata - FA [47]. Despite its accurate and successful application in some areas, its major drawback is its probabilistic approach. The probabilistic approach brings the nondeterminism into the Automata. Accordingly, this leads to a) complication and ambiguities at the time of construction and b) at the time of classification the non-determinism leads to unstable results. This instability is the result of the loss of information due to the low probability rules.

Name: FA Construction
Input: Set of association rules: ruleSet
Output: Set of Automata
1: Read rules one by one from the ruleSet until
the ruleset is Empty
2: if no conflict (ruleSet, set FA) then
3: insert into Automata
4: else Add rule to conflictRuleSet
6: end if
7: Update ruleSet = conflictRuleSet
8: Call FA Construction with updated ruleset
Algorithm 1: Generation Of Nondeterministic Finite

<u>10<sup>th</sup> June 2016. Vol.88. No.1</u>

© 2005 - 2016 JATIT & LLS. All rights reserved

JATIT
E-ISSN: 1817-3195

ISSN: 1992-8645	www.jatit.org	E-ISSN

# Automata (NFA) From Class Association Rules: Cars

In [48], a variation of Aho-Corasick (AC) String Matching algorithm [2] was used for the Non-Exact Classification. Aho-Corasick has the features needed to deal with a very small fraction of classification which was utilized by the authors in non-exact matching. They applied it to the Electrocardiogram (ECG) identification and classification with an accuracy rate of 99.67%.

The ACA is a Deterministic Finite Automatabased algorithm and its running time is linear in the size of text for string matching. ACA is divided into two phases namely (a) FA\_Construction phase that generates Automata from the CAR generated by the Apriori Algorithm and (b) Classification phase that classifies the test data. The algorithms for both phases are shown in Algorithm 1 and 2 respectively.

# Name: Classification Input: Set of Automata: testData Output: Classified Dataset

1: read rule one by one until the testData is Empty 2: Set Accuracy to 0.0
3: Set LEVEL to the total number of Automata
4: Find the Automata that Maximizes the ratio in a Level wise fashion and assign the final state as a class label
5: Read next pair
6: ratio = (hit/totalStates) \* (hit + totalStates) \* 2
7: If no matched Automata is found? Assign the default class label

Algorithm 2: Classification Of Test Dataset Name: ACA Classifier

# 3. MINIMIZING LOSS OF INFORMATION IN ACA

ACA does not use the traditional Pruning techniques that help the ACA to deal with the majority of rules generated by the Apriori Algorithms. The rationale behind keeping the majority of rules is that insignificant rules are already ignored by setting the minimum threshold i.e. support and confidence. There are two questions arise when keeping the maximum number of rules in the classifiers. a) How to deal with repeated rules? And, b) how to handle conflicting rules? The default nature of Automata handles the repetition. Whenever it encounters similar rules, it rules automatically absorbs the into single

Automata. The advantages of absorbing are twofold. First, the size of Automata remains small as its only increment the counts for the repeated attributes. Secondly, the rule that contains information is not a loss but absorbed in the Automata. To understand the concept, let us take a look at rule numbers 2 and 3 of Table 1. The resultant Automata is shown in Figure 1. In all other techniques, the pruning phase will choose one of these rules. While the ACA keeps all information inside the Automata. It, therefore, ensures that meaningful information will not be lost. In order to avoid the data overfitting, 10 fold cross validations are used which reduce the chance of data overfitting.

In classification phase, we must check these specialized FA first in order to avoid the wrong classification. The logic is that the low-level FA is more generalized due to the absorption of many rules as compared to the special Automata. On the contrary, the higher level Automata represents very few rules and sometimes may be only one rule.

The question might arise that if they cause a conflict at the time of FA construction, how could they be absorbed by those conflicting FA at the time of classification. The rationale is that during the FA construction phase, the conflict arose at the class label and now the test instance is without the class



Figure 1: The ACA Example Of The Capability Of Absorbing Similar Rules

Now, the information loss minimization is discussed in detail on how the Automata can help to reduce the loss of Information. Two properties of the ACA are discussed in detail on how a conflict is

<u>10<sup>th</sup> June 2016. Vol.88. No.1</u>

© 2005 - 2016 JATIT & LLS. All rights reserved.



WW

www.jatit.org

E-ISSN: 1817-3195

resolved in Automata and What is the role of the Level Wise search during the classification.

ISSN: 1992-8645

Conflict Resolution: Conflict refers to a situation where it is indecisive to choose the class label for test data under normal circumstance i.e.  $X \rightarrow C0$ and  $X \rightarrow C1$ . In order to handle conflicts, every rule starts the comparison with the first FA and will try to absorb that rule. If the rule is conflicting with the comparing FA, the Algorithm will check it with the next level FA and so on until either one of the existing Automata absorbs the incoming rule or new Automata is built for that rule. Similarly, the weight of the automata will increase by every absorbing rule. During classification, the conflict will be resolved using the weight of the Automata i.e. the Automata with a higher number of the absorbing rule that will give preference over the less number.

Level wise Automata: The Automata construction in a level-wise manner helps to float the rare rule (Specialized Rule) into a higher level. The rare rules are those with low support and generally, these rules appear at the end of the CARs' list (Weka sorts the rules based on confidence (Descending), size (Ascending), and Support (Descending). Let us take a look at Table 1 that shows the partial CARs for Contact Lens Datasets from UCI [49]. In this example, we will highlight the reasons behind the high accuracy of ACA, as well as the importance level of the wise Automata construction in terms of information loss minimization. The datasets were converted to a simple form of a consecutive integer as values and character as a column heading for easy reference and understanding.

Table 1: Cars Generated Using Weka 3.7.13 For Contact\_Lens Dataset

S No	А	В	С	D	Class
1	null	null	null	0	C0
2	null	0	null	0	C0
3	null	1	null	0	C0
			•		
			•		
			•		
9	null	0	1	1	C2
			•		
			•		
			•		
16	0	null	1	1	C2
			•		
			•		
			•		

22	1	0	null	0	C0
23	1	1	1	null	C1
24	1	1	null	0	C0
25	1	null	0	0	C0
26	1	null	1	0	C0
27	2	0	0	null	C1
28	2	0	null	0	C0
29	2	1	1	null	C2
30	2	1	null	0	C0
31	2	null	0	0	C0
32	2	null	1	0	C0
33	1	1	1	1	<u>C</u> 0
34	2	0	0	1	<u>C</u> 0
35	2	1	1	1	C0

In Table 1, Rule Numbers 33 to 35 are rare rules because they appear in only one transaction and therefore, most of the AC techniques will prune them either due to the low support or in the conflict with the high support rules. The ACA will not only consider these rules, but also it will oat them automatically to the higher level FA. The reason is that they are conflicting with rule numbers 9 and 16 where they both leads to different classes at the same value of last variable i.e "D", and therefore will make a new automaton. Now, the FA constructed at the \end" of list will obviously have less number of further rules to absorb; hence, these later FA will be considering as "special" due to few number of \special rules". When it come to the classification phase, we must check these specialized FA first in order to avoid the wrong classification. The logic is that the lowlevel FA is more generalized due to the absorption of many rules as compared to the special Automata. On the contrary, the higher-level Automata represent very few rules and sometimes may be only one rule.

The question might arise that if they cause a conflict at the time of FA construction, how could they be absorbed by those conflicting FA at the time of classification. The rationale is that during the FA construction phase, the conflict arose at the class label and now the test instance is without the class label; as a result, this time, there will be no conflict and the test instance may result in a 100% percent match with the wrong class label. Therefore, we need to check the specialized FA (higher Level) first and move to the lower level FA in a sequence.

During classification, we have two possibilities. If there is only one automaton in the model, then the procedure is simple and all the test instances will test against a single automaton. In the

<u>10<sup>th</sup> June 2016. Vol.88. No.1</u> © 2005 - 2016 JATIT & LLS. All rights reserved<sup>.</sup> JATIT

ISSN: 1992-8645	www.jatit.org	E-ISSN: 1817-3195

case of multiple Automata, the classification starts from a higher level of the automaton.

The classification of test instances is based on the "attribute-value" pairs, similar to the one used in FA construction. The algorithm verifies both attributes and its value to a single state. If both matches, the process then moves to the next state and reads the next pair from the test instance. Every match in the Automata increments "hit", while mismatches incur an increment of a "miss" variable. At the time of classification, the algorithm checks the ratio between "hit - miss" and the highest weighted ration is considered the predicted class for the instance.

The weighted ration is defined by Eq (1). This equation gives a high weight to those Automata with a maximum number of matching states. For example, for three hits out of six states, FA will give a high ration as compared to two hits out of four states; while the average for both is 0.5. Here we focus on the maximum number of matches instead of the maximum average where five out of ten and ten out of twenty are equal using a simple average. However, Eq (1) gives preference to ten out of twenty due to the maximum numbers of correct matches. It results in a stronger classification as a consequence of high number of similarity with respect to the number of attributes. Furthermore, if the numbers of hits are the same, then it gives preference to a lesser number of mismatches; therefore, in calculating weight, Eq (2) will increase the weight of low miss-count and decrease that of the high miss-count.

$$Ration = \omega + hit\_counts \tag{1}$$
$$\omega = \left(\frac{1}{2}\right) * \left(\frac{1}{2}\right) \tag{2}$$

$$\omega = \left(\frac{1}{miss\_count}\right)^* \left(\frac{1}{totat\_attributes}\right)$$
(2)

Example: Tables 2 and 3 correspond to the lenses and dermatology contact datasets respectively. In each table, the row represents the outcome of the comparison of the test instance with one automaton. Column 1 (S No) is an identifier for each automaton while column 4 (Total) represents the total number of states in that automaton. The test instance tests against six different types of Automata are shown in Table 2 along with their hit and miss counts. Based on Eq 1, the algorithm will select O1 because it has the maximum number of hits and a lesser number of misses as compared to Q3. In this case, the simple average will also select Q1 because the average is the highest among all; despite the issue that a single hit of Q5 is given preference over two hits of Q6. Now consider Table 3: the

simple average will now select Q1 as a result of the highest average and will ignore Q2 where 14 attributes match out of 16. However, ACA will choose Q2 because it gives the highest ration.

Table 2: Partial FA From The Contact Lens Dataset For Weighted Class Label Comparison (Total No. Of Attributes: 4)

S				Eq	
No	Hit	Miss	Total	(I)	Average
Q1	8	1	9	8.1	0.89
Q2	14	2	16	14.0	0.88
Q3	2	1	3	2.1	0.67
Q4	3	2	5	3.0	0.60
Q5	4	3	7	4.0	0.57
Q6	5	4	9	5.0	0.56

Table 3: Partial FA From Dermatology Dataset For Weighted Class Label Comparison (Total No. Of Attributes: 16)

S No	Hit	Miss	Total	Eq (I)	Average
Q1	3	1	4	3.2	0.75
Q2	2	1	3	2.2	0.67
Q3	3	2	5	3.1	0.60
Q4	2	2	4	2.1	0.50
Q5	1	1	2	1.2	0.50
Q6	2	3	5	2.1	0.40

#### 4. METHODS AND MATERIALS

This section provides the experimental results of the ACA and its comparison with the existing state-of-the-art classifiers. The ACA is implemented using the Java version 1.7 of Windows 7 running on 64bit core i3 2.30GHz machine with 8GB of memory.

For the purpose of comparison, we randomly choose ten datasets from the most commonly appearing datasets in literature, including: [9,11,13,15,28,32]. All datasets are available online at UCI Machine Learning Repository [49]. Description of datasets are given in Table 4. The datasets were collected from both continuous and discrete domain. Table 4 explains the number of attributes, instances, and number of classes in each dataset.

10<sup>th</sup> June 2016. Vol.88. No.1

© 2005 - 2016 JATIT & LLS. All rights reserved

ISSN: 1992-8645

<u>www.jatit.org</u>

Table 4: Description Of Datasets

Dataset	Attributes	Instances	Classes
contact-lenses	4	24	3
iris	4	150	3
iris2D	4	150	3
weathernominal	4	14	2
weathernumeric	4	14	2
hayes-roth	5	160	3
tae	5	151	3
car	6	1728	4
liver-disorders	7	345	2
cmc	9	1473	3

The rest of the results are generated using Weka 3.7.10. In order to make the regeneration of result easier, all experiments were conducted with the default parameters setting. Furthermore, as the classification techniques need discrete data to work upon, therefore, the numeric datasets were discretized using the unsupervised discretizer of Weka. The ACA is not using the pruning techniques, therefore, keeping the maximum number of rules in consideration. That helps to avoid the loss of Information.

#### 5. RESULTS AND DISCUSSION

ACA was compared with three different classes of classification algorithms. The first is obviously based on the association rules, so as to give a proper comparison with its own family members, i.e. CBA and LAC. The second set is rule-based algorithms. This class of algorithms is the most similar in nature with AC as they are based on general rules, while ACA is based on the Class Association Rules. Algorithms in this class are Jrip (Ripper), J48 (C4.5), OneR, ZeroR and PART respectively. Finally, the Average One Dependence Estimator (AODE) is selected from Bayesian Family which considers a family of the most prominent classifiers with reasonably high accuracy.

The ACA was evaluated using two different measuring techniques; namely, the Accuracy and weighted F-Measure among Accuracy. Precision. recall, and F measure. The reason for selecting these two techniques is that accuracy evaluates the correctness of algorithm as it gives True Positive (TP) ratio of all classes. F-Measure, on the other hand, is the Harmonic mean of precision and recall. Precision means that techniques returned mostly exact results while recall says that the result retrieved by the technique is complete [21]. Furthermore, F Measure can be extended to recall or precision evaluation by using the  $F_{\beta}$  measure instead of the F measure where, if the value of  $\beta < 0$  then it is precision sensitive, otherwise it is recall sensitive. In this experiment, all the results were generated using the ten-fold cross-validation.

Tables 5 and 6 represent the results of Accuracy and F-Measure respectively. The Accuracy is calculated using equation 3 while equation 4 computes the Weighted F Measure. In both equations, N represents the total number of records and m shows the total number of distinct class labels.

$$Acc = \frac{100}{N} \left( \sum_{i=1}^{m} TP(Class_i) \right)$$
(3)

$$\omega f = \frac{1}{N} \left( \sum_{i=1}^{m} (f(class_i)^*(count(class_i))) \right)$$
(4)

Where  $f(class_i)$  is the standard f-measure formula which is given in equation (5)

$$f(i) = 2* \left( \frac{precision_i * recall_i}{precision_i + recall_i} \right)$$
(5)

SNo	Dataset	AODE	CBA	Jrip	OneR	PART	ZeroR	J48	LAC	ACA
1	car	92.5	77.8	86.5	70.0	95.8	70.0	92.4	70.0	82.0
2	contact-lenses	68.3	68.3	75.0	71.7	81.7	68.3	81.7	62.5	85.6
3	hayes-roth	76.5	37.9	82.5	44.0	74.1	37.9	72.7	76.5	68.2
4	iris	94.7	66.0	93.3	96.0	95.3	33.3	96.0	96.7	96.9
5	iris2D	96.7	66.0	93.3	96.0	92.7	33.3	96.0	94.0	98.2
6	liver-disorders	64.1	58.0	59.4	55.4	62.4	58.0	61.2	66.1	61.8
7	tae	56.2	34.4	56.3	49.7	47.0	34.4	49.7	58.3	64.3
8	weathernominal	55.0	55.0	70.0	45.0	60.0	70.0	55.0	71.4	85.0
9	weathernumeric	40.0	70.0	50.0	35.0	55.0	70.0	55.0	35.7	80.0
10	cmc	53.0	42.7	48.3	47.5	48.0	42.7	49.3	51.4	62.2

Table 5: Accuracy Comparison With 8 Algorithms Over 10 Different Dataset

<u>10<sup>th</sup> June 2016. Vol.88. No.1</u>

© 2005 - 2016 JATIT & LLS. All rights reserved

ISSN: 1992-8645	www.jatit.org	E-ISSN: 1817-3195

S No	Dataset	AODE	CBA	Jrip	OneR	PART	ZeroR	J48	LAC	ACA
1	car	0.9	0.8	0.9	0.6	1.0	0.6	0.9	0.6	0.7
2	contact-lenses	0.7	0.7	0.7	0.7	0.8	0.6	0.8	0.5	0.8
3	hayes-roth	0.8	0.2	0.8	0.4	0.7	0.2	0.7	0.8	0.5
4	iris	1.0	0.6	0.9	1.0	1.0	0.2	1.0	0.9	1.0
5	iris2D	1.0	0.6	0.9	1.0	0.9	0.2	1.0	1.0	1.0
6	liver-disorders	0.6	0.4	0.6	0.5	0.6	0.4	0.6	0.6	0.5
7	tae	0.6	0.2	0.6	0.5	0.5	0.2	0.5	0.6	0.5
8	weathernominal	0.5	0.5	0.7	0.4	0.6	0.6	0.5	0.7	0.8
9	weathernumeric	0.4	0.7	0.5	0.3	0.5	0.6	0.5	0.3	0.8
10	cmc	0.5	0.3	0.4	0.4	0.5	0.3	0.5	0.5	0.5

Table 6: F Measure Comparison Of ACA With 8 Different Algorithms Over 10 Dataset

The results in Table 4 show that ACA is far better than all the rule-based algorithms although the Jrip, PART and LAC perform better in individual datasets; namely, the hyes-roth, car and liver-disorder respectively. Overall, ACA performed better in seven out of ten datasets with respect to accuracy. The new algorithm clearly outperforms AODE, CBA, ZeroR, OneR and J48. The reason for low performance of other three datasets might be the limitation of CARs generation on our system while the reason for the high

accuracy of ACA, over other techniques, is the replacement of pruning with merging. The ACA avoid pruning while the default nature of the ACA merges the similar rules which result in the smaller rule set.

Figure 2 also highlights that ACA is very good with both continuous dataset and discrete datasets with few exceptions. A more precise conversion from continuous to discrete conversion may help to improve the performance for the hayes-roth and liver-disorder dataset as well.



Figure 2: Accuracy Comparison Of ACA With State-Of-The-Art Classification Techniques

Next, the F Measure in Table 6 shows the realistic behaviour of ACA where the results are not stable like accuracy. The reason is the calculation differences in the two measurements. F Measure takes both precision and recall in account while accuracy only considers the true-positive and true-negative results, therefore, in general, when the accuracy is low, the F Measure becomes worse and it is shown in Table 6 in all results with few exceptions.

# 6. CONCLUSION AND FUTURE WORK

The results of ACA show that an increased number of rules reduce the loss of information as well positively affect accuracy, while reducing the number of rules can cause information loss that can reduce the accuracy. The previous techniques use pruning for performance improvement while analysis showed that ACA can handle a large set of rules efficiently with a high number of accuracy and F Measure. The current approach has a limitation in dealing with continuous data. In this context, the

<u>10<sup>th</sup> June 2016. Vol.88. No.1</u>

© 2005 - 2016 JATIT & LLS. All rights reserved

ISSN: 1992-8645	www.jatit.org	E-ISSN: 1817-3195

concept of fuzziness can be utilized in ACA. The algorithm has the capability by which to tune the accuracy. The ACA merges the rules based on the structure similarity; therefore, a new weighting criterion assigns weight to individual rules in Automata that might increase the accuracy. Similarly, a probabilistic approach The results in Table 4 show that ACA is far better than all the rule-based algorithms although the Jrip, PART, and LAC perform better in individual datasets; namely, the hyes-roth, car and liver-disorder respectively. Overall, ACA performed better in seven out of ten datasets with respect to accuracy.

The new algorithm clearly outperforms AODE, CBA, ZeroR, OneR and J48. Figure 2 highlights the comparison graphically. The reason for low performance of other three datasets might be the limitation of CARs generation on our may also help in reducing the number of rules with loss-less merging that can further improve the efficiency of the algorithm.

# 7. ACKNOWLEDGEMENT

This work is supported by the Fundamental Research Grant Scheme (FRGS) of Ministry of Higher Education (MOHE), Malaysia with reference to FRGS/1/2014/ICT01/UTM/02/2 (Vot: 4F431). Universiti Teknologi Malaysia (UTM) is gratefully acknowledged for its Research University Grant (Vot: 11H92).

# REFERENCES

- [1] O. Duda, P. E. Hart, and D. G. Stork. Pattern classification. John Wiley & Sons, San Jose, California, 2012.
- [2] T. Hastie, R. Tibshirani, and J. J. H. Friedman. The elements of statistical learning, volume 1. Springer New York, 2001.
- [3] C. Cortes and V. Vapnik. Support-vector networks. Machine learning, 20(3):273-297, 1995.
- [4] J. Hong, I. Mozetic, Michalski, and R. S. Aq15: Incremental learning of attribute-based descriptions from examples. Machine Learning an Artificial Intelligence Approach, Tioga Publishing, Palo Alto, 1986.
- [5] J. R. Quinlan. Simplifying decision trees. International journal of man-machine studies, 27(3):221-234, 1987.
- [6] J. R. Quinlan. C4. 5: programs for machine learning, volume 1. Morgan Kaufmann, 1993.
- [7] T. Cover and P. Hart. Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1):21-27, 1967.

- [8] W. H. Bing Liu and Y. Ma. Integrating classification and association rule mining. In Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, pages 80-86, 1998.
- [9] J. Han. CPAR: Classification based on predictive association rules. In Proceedings of the third SIAM inter- national conference on data mining, volume 3, pages 331-335. Siam, 2003.
- [10] J. Li, H. Shen, and R. Topor. Mining the smallest association rule set for predictions. In Proceedings IEEE International Conference on Data Mining, pages 361-368. IEEE, 2001.
- [11] W. Li, J. Han, and J. Pei. Cmar: Accurate and efficient classification based on multiple class-association rules. In Proceedings IEEE International Conference on Data Mining, pages 369-376. IEEE, 2001.
- [12] F. Thabtah. Rules pruning in associative classification mining. In Proceedings of the IBIMA Conference, pages 7-15. Citeseer, 2005.
- [13] F. Thabtah, P. Cowling, and Y. Peng. Mcar: multi- class classification based on association rule. In Computer Systems and Applications, 2005. The 3rd ACS/IEEE International Conference on, page 33. IEEE, 2005.
- [14] K. Ali, S. Manganaris, and R. Srikant. Partial classification using association rules. In Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining (KDD\_S97), pages 115-118, 1997.
- [15] N. Abdelhamid, A. Ayesh, F. Thabtah, S. Ahmadi, and W. Hadi. Mac: A multiclass associative classification algorithm. Journal of Information & Knowledge Management, 11(02):1-10, 2012.
- [16] E. Baralis, S. Chiusano, and P. Garza. A lazy approach to associative classification. IEEE Transactions on Knowledge and Data Engineering, 20(2):156-171, 2008.
- [17] E. Baralis and P. Garza. A lazy approach to pruning classification rules. In Proceedings IEEE International Conference on Data Mining, pages 35-42. IEEE, 2002.
- [18] J. Nahar, T. Imam, K. S. Tickle, and Y. P. P. Chen. Computational intelligence for heart disease diagnosis: A medical knowledge driven approach. Expert Systems with Applications, 40(1):96-104, 2013.
- [19] J. Friedman. A recursive partitioning decision rule for nonparametric classification. Computers, IEEE Trans- actions on, C-26(4):404-408, 1977.

<u>10<sup>th</sup> June 2016. Vol.88. No.1</u>

© 2005 - 2016 JATIT & LLS. All rights reserved

- [20] S. L. Wang, Y. Fang, and J. Fang. A simple but robust complex disease classification method using virtual sample template. In D. S. Huang, P. Gupta, L. Wang, and M. Gromiha, editors, Emerging Intelligent Computing Technology and Applications, volume 375 of Communications in Computer and Information Science, pages 73-80. Springer Berlin Heidelberg, 2013.
- [21] V. Alabau, A. Sanchis, and F. Casacuberta. Improving on-line handwritten recognition in interactive machine translation. Pattern Recognition, 47(3):1217-1228, 2014.
- [22] Graves, A. Mohamed, and G. E. Hinton. Speech recognition with deep recurrent neural networks. CoRR, abs/1303.5778, 2013.
- [23] P. M. Jain, D. A. Gawande, and P. L. Gautam. Image mining for image retrieval using hierarchical k-means algorithm. International Journal of Research in Computer Engineering & Electronics, 2(6):1-6, 2014.
- [24] E. Mavridou, D. D. Kehagias, K. Kalogirou, and D. Tzovaras. An agent-oriented data mining frame- work for mass customization in the automotive industry. In IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, volume 3, pages 575-578. IEEE, 2008.
- [25] Y. Ye, Q. Jiang, and W. Zhuang. Associative classification and post-processing techniques used for malware detection. In 2nd International Conference on Anticounterfeiting, Security and Identification, pages 276-279. IEEE, 2008.
- [26] Y. Ye, T. Li, Q. Jiang, Z. Han, and L. Wan. Intelligent \_le scoring system for malware detection from the gray list. In Proceedings of the 15th ACM SIGKDD inter- national conference on Knowledge discovery and data mining, pages 1385-1394. ACM, 2009.
- [27] G. Costa, M. Guarascio, G. Manco, R. Ortale, and E. Ritacco. Rule learning with probabilistic smoothing. In Data Warehousing and Knowledge Discovery, pages 428-440. Springer, 2009.
- [28] F. Thabtah, Q. Mahmood, L. McCluskey, and H. Ab- del Jaber. A new classification based on association algorithm. Journal of Information & Knowledge Management, 9(01):55-64, 2010.
- [29] J. Li, R. Topor, and H. Shen. Construct robust rule sets for classification. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 564-569. ACM, 2002.

- [30] M. L. Antonie, O. R. Zaiane, and A. Coman. Associa- tive classifiers for medical images. In Mining Multimedia and Complex Data, pages 68-83. Springer, 2003.
- [31] L. Bing, H. Wynne, and M. Yiming. Integrating classification and association rule mining. In Proceedings of the Fourth International Conference On Knowledge Discovery And Data Mining, pages 80-86, 1998.
- [32] B. Liu, Y. Ma, and C. K. Wong. Improving an association rule based classifier. In Principles of Data Mining and Knowledge Discovery, pages 504-509. Springer, 2000.
- [33] K. Wang, S. Zhou, and Y. He. Growing decision trees on support-less association rules. In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 265-269. ACM, 2000.
- [34] B. Depaire, K. Vanhoof, and G. Wets. Arubas: an association rule based similarity framework for associative classifiers. In IEEE International Conference on Data Mining Workshops, pages 692-699. IEEE, 2008.
- [35] H. Witten and E. Frank. Data Mining: Practical ma- chine learning tools and techniques. Elsevier, 3rd edition, 2011.
- [36] E. Baralis, S. Chiusano, and P. Garza. On support thresholds in associative classification. In Proceedings of the 2004 ACM symposium on Applied computing, pages 553-558. ACM, 2004.
- [37] F. A. Thabtah, P. Cowling, and Y. Peng. Mmac: A new multi-class, multi-label associative classification approach. In Fourth IEEE International Conference on Data Mining, pages 217-224. IEEE, 2004.
- [38] G. Kundu, M. M. Islam, S. Munir, and M. F. Bari. Acn: An associative classifier with negative rules. In 11th IEEE International Conference on Computational Science and Engineering, pages 369-375. IEEE, 2008.
- [39] E. Roche and Y. Schabes. Deterministic partof-speech tagging with finite-state transducers. Computational linguistics, 21(2):227-253, 1995.
- [40] P. Hingston. Using finite state automata for sequence mining. In O. Michael, editor, Proceedings of the Twenty-Fifth Australasian Computer Science Conference (ACSC2002) on Research and Practice in IT, volume 4, pages 105-110. Australian Computer Society, Inc., 2002.

<u>10<sup>th</sup> June 2016. Vol.88. No.1</u>

© 2005 - 2016 JATIT & LLS. All rights reserved

		JAIII			
ISSN: 1992-8645	www.jatit.org	E-ISSN: 1817-3195			
[41] V. Aho and M.	J. Corasick. Efficient string				

- [41] V. Aho and M. J. Corasick. Efficient string matching: an aid to bibliographic search. Communications of the ACM, 18(6):333-340, 1975.
- [42] Y. Fan, H. Zhang, J. Liu, and D. Xu. An efficient parallel string matching algorithm based on DFA. In Trustworthy Computing and Services, pages 349-356. Springer, 2013.
- [43] M. Mohri. Matching patterns of an automaton. In Combinatorial Pattern Matching, pages 286-297. Springer, 1995.
- [44] B. J. Oommen and E. V. de St Croix. Graph partitioning using learning automata. IEEE Transactions on Computers, 45(2):195-208, 1996.
- [45] P. Sastry and M. Thathachar. Learning automata algorithms for pattern classification. In Sadhana (Academy Proceedings in Engineering Sciences), volume 24, pages 261-292. Indian Academy of Sciences, 1999.
- [46] S. H. Zahiri. Learning automata based classifier. Pattern Recognition Letters, 29(1):40-48, 2008.
- [47] W. Homenda and W. Pedrycz. Finite automata with imperfect information as classification tools. In Computational Collective Intelligence. Technologies and Applications, pages 465-474. Springer, 2012.
- [48] K. Tseng, F.-F. Zeng, H. N. Huang, Y. Liu, J. S. Pan, W. Ip, and C. Wu. A new non-exact aho-corasick framework for ecg classification. ACM SIGARCH Computer Architecture News, 41(2):41-46, 2013.
- [49] Bache and M. Lichman. UCI: machine learning repository, 2013.
- [50] J. Han, M. Kamber, and J. Pei. Data mining: concepts and techniques. Morgan kaufmann, 3rd edition, 2012.
- [51] J. E. Hopcroft. Introduction to Automata Theory, Languages, and Computation. Pearson Education India, India, 3rd edition, 2008.
- [52] Mohammad Abrar, Alex Tze Hiang Sim (2016), Journal Teknologi (in Press).