

Automata Bases Associative Classification (AAC) for Data Mining

Mohammad Abrar¹⁺, Alex Tze Hiang Sim¹ and Jee Mei Hee²

¹ Faculty of Computing, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia

² Faculty of Education, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia

Abstract. The study on the use of association rules for the purpose of classification gave rise to a new field known as Associative Classification (AC). The process used to generate association rules is exponential by nature; thus in AC, researcher focused on the reduction of redundant rules via rules pruning and rules ranking techniques. The removal of rules however could negatively affect accuracy. In this paper, we radically store most of the rules in a condensed form utilizing automata. The automata offsets critical need for rules pruning and ranking. Our new structure is used for classification. Experimental results show that the accuracy of our automata based technique is significantly improved compare to the existing state-of-the-art algorithms which includes J48, AODE, BayesNet and FT etc. The analysis also shows that our automata based associative classification technique is efficient by means of computational time and space utilization.

Keywords: associative classification, automata, association rules, data mining, machine learning.

1. Introduction

Recent studies of association rules (AR) show promising use of AR for classification. After its introduction in 1997 [1], algorithms developed in this family include CBA[1,2], CMAR[3],MCAR[4], CPAR[5], MAC[6] and almost all consist of three basic steps, specifically: (I) To generate class-association rules, (II) To prune and rank rules, and (III) To build and evaluate the classifier.

In this paper, we argue for an efficient structure to incorporate association rules into a classifier with experimental results comparable to state-of-the-art algorithms includes J48, AODE, BayesNet, Jrip and FT etc. The rest of the paper is organised as follows: In Section 2, we report related work and needed information to understand and appreciate AC; in Section 3 our main algorithm and logic in merging and building an efficient structure. This is followed by experimental results and analysis of the algorithm in Section 4. The last Section 5 concludes the paper with future research directions.

2. Literature Review

Associative Classification [1] is an integration of two major techniques in data mining – Association Rules Mining and Classification. In 1998 [7] showed that such an integration is able to increase accuracy of the existing classifiers. In these, it is a stronger classifier as compared to traditional algorithms. A number of researchers have since, conducted various research on associative classifier. Among these, [8] reviewed and summarised that association rules type of classification can produce more accurate classifier compare to the traditional approaches such as decision tree and rule induction. In addition, author highlighted a number of research directions for associative classification which includes pruning for rules while maintaining the completeness of solutions. The latter is an important issue addressed in this research.

Importantly, rule pruning is another critical research topic as association rules mining typically output huge number of rules. Baralis [9] proposed to prune the association rules discovered by rule ranking –

⁺ Corresponding author. Tel.: +(60) 112 8688186.
E-mail address: muhammadabrar78@gmail.com.

Comparing the confidence and support values of the longest rules (term as compact rules). Whereas,[10] highlighted five popular rule pruning approaches. (I) Chi-squared test to test if a rule antecedent is correlated with its consequence based on statistics (II) Redundant rule pruning based on confidence value (III) Redundant rule pruning based on coverage value (IV) error estimation based on estimated error of a new rule in order to replace the existing rule (V) Redundant rule pruning based on comparison between the support and confidence values. Although the smaller number of rules in a classifier is advantageous in the context of speed but it is less appreciated for accuracy with an additional computational overhead to the classifier. Associative Classification Using Automata

In this article, we propose a unique algorithm that constructs Finite Automata (FA) from CARs. The structure could later be scored for classifying (test) data. Algorithm 1 summarizes the procedure of FA construction from CARs for AAC.

2.1. Algorithm

In FA construction, the key part is that of transition function (δ). It defines the rules of movement from one state to another. The general form of transition function is $\delta = Q \times \Sigma \rightarrow Q$ which means that from any state (from Q) using any input symbol (from Σ) we can move to any state (in Q). CARs in Table 1 represent δ for AAC.

Algorithm 1: Creation of Finite Automata from Class Association Rules

Name: FA Construction

Data: Set of (class-) association rules ruleSet

Result: Set of Automata

```

1: create Automata with rule-1 from ruleSet and increment Level
2: while ruleSet,  $\varphi$  do
3:   read rule one by one
4:   if no conflict (ruleSet, set FA) then
5:     insert into Automata
6:   else
7:     add rule to conflictRuleSet
8:   end if
9: end while
10: Update ruleSet = conflictRuleSet
11: Call FA Construction with updated ruleSet

```

2.2. Mapping of CARs

In this section and throughout the rest of the paper, we will refer to Table 1 as our working example. Table 1 represents CARs for IRIS2D dataset from UCI repository [11]. The dataset is discretised using WEKA 3.7.10 unsupervised discretiser. The CARs are generated using Weka’s Apriori Algorithms with default parameter setting. There are a total of 9 rules; each row represents one rule. Every row consists of, specifically: (i) a rule number (Column R No), (ii) attributes (i.e. columns PL (Patel Length) and PW (Patel Width)) and (iii) class label (column Class). Each cell contains the value for an attribute. For example, “min-2.45” is the value of Patel Length for rules no. 1 and 3.

Table 1: CARs for IRIS2D dataset, generated using Weka 3.7.10

R No	(PL)	(PW)	Class	R No	(PL)	(PW)	Class
1	min-2.45		Iris-Setosa - C1	6	2.45-4.75		Iris-versicolor - C3
2		min-0.8	Iris-Setosa - C1	7	2.45-4.75	0.8-1.75	Iris-versicolor - C3
3	min-2.45	min-0.8	Iris-Setosa - C1	8		0.8-1.75	Iris-versicolor - C3
4		1.75-max	Iris-Virginica - C2	9	4.75-max		Iris-Virginica - C2
5	4.75-max	1.75-max	Iris-Virginica - C2				

In order to represent Table 1 as FA, we performed a following mapping: The CARs are fed to the algorithm as δ for $FA = \{Q, \Sigma, \delta, q_0, F\}$ where, specifically: Q is the union of set of attributes and Distinct Class labels from datasets; Σ is the collection of distinct values of all attributes; q_0 is the start state and can be any attribute from set of $Q - F$. Finally F , set of final states, is the set of distinct class labels. The following illustration explains how CARs are incorporated into an automata.

Illustration 1: This example shows how Table 1 is represented in our AAC.

All rules in Table 1 represent δ . These rules, actually, provide the mechanism of movement between different states of FA.

The set of states:

$Q = \{PL, PW, C1, C2 \text{ and } C3\}$ (Collection of all attributes and distinct class labels).

The input symbols:

$\Sigma = \{min - 2.45, 2.45 - 4.75, 4.75 - max, min - 0.8, 0.8 - 1.75, 1.75 - max\}$ (set of distinct values of all attributes.)

q_0 can be any attribute that starts a rule; e.g. for rule 1 & 3 q_0 is 'PL' (Pedal Length), while for rules 2 & 4 etc., it is 'PW' (Pedal Width).

Lastly, the set of final states:

$F = \{C1, C2, C3\}$ (distinct class labels).

Algorithm 1 adds rules to the existing FAs one by one. In the case of distinct rule, it will become a candidate for a new automaton. The process continues until there are no more rules in the rule set. The step wise process is illustrated in Figures 1-3. When the algorithm reads rule 2 from Table 1, it starts from PW (anew) which is different from the start state of FA0, thus resulted in a distinct rule in FA (denoted as FA1). This is shown in Figure 2. After reading rule 3 from Table 1, the algorithm will compare it with FA0. Since there is no distinction, FA0 will absorb rule 3 and the structure of FA0 changed to its revised form as shown in Figure 3.

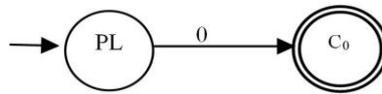


Fig. 1: FA0 after reading rule 1 from Table 1.

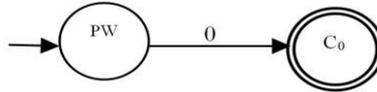


Fig. 2: FA1 after reading rule 2 from Table 1.

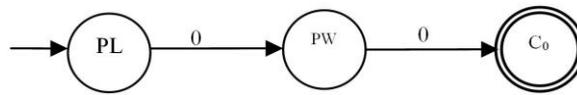


Fig. 3: FA0 after reading rule 3 from Table 1.

3. Results and Discussion

This section provides the experimental results of AAC and its comparison with the existing state-of-the-art classifiers. AAC is implemented using Java version 1.7 on Windows 8.1 running over 64bit core i5 2.30GHz machine with 8GB of memory. The results for other classifiers are generated using Weka 3.7.10 on the same machine. In order to make the regeneration of result easier, all experiments were conducted with default parameters. Further, as classification techniques need discrete data to work upon, therefore, the numeric dataset were discretised using unsupervised discretiser of Weka.

For the purpose of comparison, we randomly choose three datasets from the most commonly appearing dataset in literature, including: [3, 4, 5, 6]. All datasets are available online at UCI Machine Learning Repository [11].

Table 2 shows a comparison of AAC and Naive Bayes family of classifiers. The results based on 10-fold cross validation show that AAC outperforms the Naive Bayesian all family members in two datasets with a reasonable difference. The result also shows that in most cases the NB based algorithms results in same accuracy.

Table 2: AAC Comparison with NB Classifiers

S No	Classifier	balance-scale	contact-lenses	iris2D
1	AAC	73.61	91.67	97.78
2	AODE	89.45	68.33	96.67
3	BayesNet	91.36	71.67	96.67
4	NaiveBayes	91.36	71.67	96.67
5	NaiveBayes_Updatable	91.36	71.67	96.67

Similarly, the AAC comparison with tree based algorithm is shown in Table 3. AAC outperform the tree based algorithms in two datasets, namely; a) Contact Lens and b) IRIS2D.

Table 3: AAC Comparison with Tree Based Classifiers

S No	Classifier	balance-scale	contact-lenses	iris2D
1	AAC	73.61	91.67	97.78
2	ID3	38.09	71.67	94.00
3	J48	64.48	81.67	96.00
4	FT	91.99	68.33	96.67
5	Jrip	70.55	75.00	93.33

Our new algorithm outperforms some of the classifiers. The reason for a lower performance of balance scale dataset could be due to the limitation of CARs generation on our system while the reason for the high accuracy of AAC, over other techniques, is replacement of pruning with merging. Fundamentally, our AAC technique avoids pruning it merges the similar rules which results in a smaller rule set. That helps to improve the efficiency as well as accuracy.

4. Conclusion

In this research, we have designed an automata based associative classifier (termed as AAC) based on our hypothesis that a reduced number of class-association rules or production rules in most other types of classifiers, reduce the accuracy for classification. Through experiments, we show that it is possible to obtain comparable even better results without removing the rules. The use of automata is appropriate to record and summarise these rules. It preserves both accuracy without much computational overhead.

5. Acknowledgment

This research is supported by research grants FRGS (vot no: 4F441), GUP Tier 1 (11H92). Authors also acknowledge the support provided by Ministry of Higher Education (MOHE), Research Management Centre (RMC) and Universiti Teknologi Malaysia (UTM).

6. References

- [1] Ali, K., Manganaris, S., Srikant, R., 1997. Partial classification using association rules , 115–118.
- [2] Bing, L., Wynne, H., Yiming, M., 1998. Integrating classification and association rule mining , 80–86.
- [3] Li, W., Han, J., Pei, J., 2001b. Cmar: Accurate and efficient classification based on multiple class-association rules , 369–376.
- [4] Thabtah, F., Cowling, P., Peng, Y., 2005. Mcar: multi-class classification based on association rule , 33.
- [5] Han, J., 2003. Cpar: Classification based on predictive association rules 3, 331–335.
- [6] Abdelhamid, N., Ayesh, A., Thabtah, F., Ahmadi, S., Hadi, W., 2012. Mac: A multiclass associative classification algorithm. *Journal of Information & Knowledge Management* 11, 1–10.
- [7] Liu, B., Hsu, W. & Ma, Y. Integrating Classification and Association Rule Mining. *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, 1998 New York. AAAI Press, 80-86.
- [8] Thabtah, F. Challenges and Interesting Research Directions in Associative Classification. *Sixth IEEE International Conference on Data Mining - Workshops (ICDMW'06)*, 2006 Hong Kong. IEEE, 785-792.
- [9] Baralis, E., Chiusano, S., Garza, P., 2008. A lazy approach to associative classification. *IEEE Transactions on Knowledge and Data Engineering* 20, 156–171.
- [10] Ye, Y., Li, T., Jiang, Q. & Wang, Y. 2010. CIMDS: Adapting Postprocessing Techniques of Associative Classification for Malware Detection. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 40, 298-307.
- [11] Bache, K., Lichman, M., 2013. UCI: machine learning repository. URL: <http://archive.ics.uci.edu/ml>.