# A New Rule Pruning Text Categorisation Method

Fadi Thabtah[1], Wa'el Hadi[1], Hussein Abu-Mansour[2], L. McCluskey[2]
MIS Dept, Philadelphia University, Jordan
[1]{whadi, ffayez}@philadelphia.edu.jo
Computing and Engineering Dept, Huddersfield University, UK
[2]{H.Y.Abu Mansour, l.MCclusky }@hud.ac.uk

***Abstract*—Associative classification integrates association rule and classification in data mining to build classifiers that are highly accurate than that of traditional classification approaches such as greedy and decision tree. However, the size of the classifiers produced by associative classification algorithms is usually large and contains insignificant rules. This may degrade the classification accuracy and increases the classification time, thus, pruning becomes an important task. In this paper, we investigate the problem of rule pruning in text categorisation and propose a new rule pruning techniques called High Precedence. Experimental results show that HP derives higher quality and more scalable classifiers than those produced by current pruning methods (lazy and database coverage). In addition, the number of rules generated by the developed pruning procedure is often less than that of lazy pruning.**

**Keywords**: Classification, Data Mining, Text categorisation, Rule pruning

## I. INTRODUCTION.

The rapid evolution in computing particularly in data collection and storage methods has lead to a dense amount of data in the different organizations' databases. This has made deriving useful information from such databases hard to achieve. Data mining can deal with such task since it utilises different intelligent algorithms for processing large data sets in order to extract useful knowledge.

Association rule discovery is one of data mining tasks that find hidden relationships among items in a transactional database. Classification is another data mining task which aims to build a model of rules called classifier from a set of labeled examples in order to use in the classification of test data sets which their class is unknown.

In recent years, a new approached called associative classification (AC) has emerged which integrates association rule and classification [9]. Several studies [9] [8] [18] [16] [15] [7] [11] provide evidence that AC is able to generate more accurate classification models than decision trees [12], and rule induction [13] [4] approaches. However, this approach normally suffers from the exponential growth of rules since AC employs association rule during the learning step where all the correlations among the items and the class are discovered in a form of if-then rules [8] [5] [7]. Moreover, some of these rules are significant and some aren't, and therefore, several pruning methods have been induce in order to cut down the number of generated rules. This is because including only the significant rules in the classifier would improve the productivity power of the classifier [2].

In this paper we investigate the rule pruning phase in AC mining in order to reduce the size of the resulting classifiers. Particularly, we develop a new rule pruning method called HP that considers the rule significant if and only if its antecedent (rule body) partially matches any of the training document keywords. This is important since we would like to keep multiple rules in the classifier for the training case which latterly are utilised in the prediction phase to improve the accuracy results.

The proposed method is implemented within a known AC algorithm called MCAR [15], and is tested against large and complex text categorisation collection called Reuters-21578 Mod Split [6].

This paper is structured as follows: AC approach and known rule pruning methods are discussed in Section 2. In Section 3, the proposed rule pruning techniques are presented. Further, the results that show the impact of pruning on the size of the

978-1-4244-7534-6/10/$26.00 ©2010 IEEE

classifiers and the prediction accuracy are demonstrated in Section 4. Finally, conclusions are given in Section 5.

## II. Associative Classification

### A. Associative Classification Problem

AC is a special case of association rule in which only the class attribute is considered in the rule's right-hand-side (consequent) [9], for example in a rule such as $X \rightarrow Y$, $Y$ must be a class attribute. We follow [16] for the definition of the AC problem. A training data set $T$ has $m$ distinct attributes $A_1, A_2, \ldots, A_m$ and $C$ is a list of class labels. The number of rows (cases) in $T$ is denoted $|T|$.

**Definition 1**: A row or a training case in $T$ can be described as a combination of attributes $A_i$ and values $a_{ij}$, plus a class denoted by $c_j$.

**Definition 2**: An attribute value can be described as a term name $A_i$ and a value $a_i$, denoted $<(A_i, a_i)>$.

**Definition 3**: An *AttributeValueSet* can be described as a set of disjoint attribute values contained in a training case, denoted $< (A_{i1}, a_{i1}), \ldots, (A_{ik}, a_{ik})>$.

**Definition 4**: A *ruleitem* $r$ is of the form $<$ *AttributeValueSet*, $c>$, where $c \in C$ is the class.

**Definition 5**: The actual occurrence (*actoccr*) of a *ruleitem* $r$ in $T$ is the number of rows (cases) in $T$ that match the *AttributeValueSet* of $r$.

**Definition 6**: The support count (*suppcount*) of *ruleitem* $r$ is the number of rows in $T$ that match $r$'s *AttributeValueSet*, and belong to the class $c$ of $r$.

**Definition 7**: The occurrence of an *AttributeValueSet* $i$ (*occatt*) in $T$ is the number of rows in $T$ that match $i$.

**Definition 8**: A *ruleitem* $r$ passes the *minsupp* threshold if $(suppcount(r)/|T|) \geq minsupp$,

**Definition 9**: A *ruleitem* $r$ passes the *minconf* threshold if $(suppcount(r)/actoccr(r)) \geq minconf$.

**Definition 10**: Any *ruleitem* $r$ that passes the *minsupp* threshold is said to be a *frequent ruleitem*.

**Definition 11**: A class association rule (CAR) is represented in the form: $(A_{i1}, a_{i1}) \wedge \ldots \wedge (A_{ik}, a_{ik}) \rightarrow c$, where the antecedent (rule body/LHS) of the rule is an *AttributeValueSet* and the consequent (RHS) is a class.

**Definition 12**: We say that a CAR $R$ partial matches a test case $t$ if R contains at least an item in its antecedent that exists in $t$.

**Definition 13**: We say that a CAR $R$ fully matches a test case $t$ if all $R$ items are in $t$.

A classifier is a mapping form $H : A \rightarrow Y$, where $A$ is the set of *AttributeValueSet* and $Y$ is the set of class labels. The main task of AC is to construct a set of rules (model) that is able to predict the classes of previously unseen data, known as the test data set, as accurately as possible. In other words, the goal is to find a classifier $h \in H$ that maximises the probability that $h(a) = y$ for each test case.

### B. Current Pruning Methods in AC

Several pruning methods have been used effectively to reduce the size of the classifiers in AC. The database coverage is a post pruning technique [9], which is usually invoked after rules have been created. If at least one case among all the cases in training data set is fully matched by the rule, the rule is inserted into the classifier and all cases covered are removed from the training data set. The rule insertion stops when either all of the rules are used or no cases are left in the training data set. The majority class among all cases left in the training is selected as default class. The default class is used in case when there are no covering rules. After this process, the first rule which has the least number of errors is identified as the cutoff rule. All the rules after this rule are not included in the final classifier since they often produce errors [9]. The database coverage method was used first by CBA [9] and then latterly by other associative algorithms, including CBA (2) [10], CMAR [8], CAAR [17], ACN [5] and Multi-label Classification based on Association Rules [7].

A pruning method that discards specific rules with less confidence values than general rules called redundant rule pruning, has been proposed in [8]. Redundant rule pruning method works as follows: Once the rule generation process is finished and rules are sorted, an evaluation step is performed to prune all rules such as $I' \rightarrow c$ from the set of generated rules, where there is some general rule $I \rightarrow c$ of a higher rank and $I \subseteq I'$. This pruning method significantly reduces the size of the resulting classifiers and minimises rules redundancy [8].

Algorithms, including [8] [1], have used redundant rule pruning. They perform such pruning immediately after a rule is inserted into the compact data structure, the CR-tree. When a rule is added to the CR-tree, a query is issued to check if the inserted rule can be pruned or some other already inserted rules in the tree can be removed.

Some AC techniques [3] [2] claim that database coverage pruning often discards some useful knowledge, as the ideal support threshold is not known in advance. Due to this, these algorithms have used a late database coverage-like approach, called lazy pruning, which discards rules that incorrectly classify training cases and keeps all others. Lazy pruning happens after rules have been created and stored, where each training case is taken in turn and the first rule in the set of ranked rules applicable to the case is assigned to it. The training case is then removed and the correctness of the class assigned to the case is checked. Once all training data have been considered, only rules that wrongly classified training data are discarded and their covered data are put into a new cycle and the process is repeated until all training data are correctly classified. The results are two levels of rules; the first level contains rules that correctly classified at least one single training case and the second level contains rules that were never used in the training phase. The main difference between lazy pruning and database coverage pruning is that the second level rules that are held in the memory by the lazy pruning are completely removed by the database coverage during rule discovery step. Furthermore, once a rule is applied to the training data, all cases covered by the rule are removed (negative and positive) by the database coverage method.

## III. The Proposed Rule Pruning Method

In this section, we discuss the proposed rule pruning method along with an example to illustrate it. Assume that the eleven rules shown in Table 1 are produced by an AC algorithm called MCAR [15] using minsupp and minconf of 20% and 40%, respectively, from the training data set shown in Table 2. Before pruning starts, the rules must be sorted in descending manner according to confidence, support, and number of items in the rule antecedent.

According to Table 1, Rule-5 is the highest ranked rule since its confidence is the largest one among the rest of the rules. Though Rule-1, Rule-6, Rule-7, Rule-9, Rule-10, and Rule-11 have the same confidence; Rule-1 is ranked higher due to its larger support. Still Rule-6, Rule-7, Rule-9, Rule-10, and Rule-11 have the same confidence and support values; but Rule-6 is ranked higher due to the fact that it has less number of values in its antecedent, and so forth.

TABLE 1: EXAMPLE OF A RULE-BASED MODEL (POTENTIAL RULES)

| Rule-id | Rule | conf | sup | Class count | Rank |
|---|---|---|---|---|---|
| 1 | real=>sport | 0.99 | 0.286 | 2 | 2 |
| 2 | madrid=>sport | 0.67 | 0.285 | 2 | 10 |
| 3 | stock=>acq | 0.75 | 0.428 | 3 | 8 |
| 4 | share=>acq | 0.67 | 0.285 | 3 | 9 |
| 5 | group=>acq | 1 | 0.285 | 3 | 1 |
| 6 | amman=>general | 0.99 | 0.285 | 2 | 3 |
| 7 | madrid & real=>sport | 0.99 | 0.285 | 2 | 6 |
| 8 | share & stock=>acq | 0.67 | 0.285 | 3 | 11 |
| 9 | group & stock=>acq | 0.99 | 0.285 | 3 | 4 |
| 10 | group & share=>acq | 0.99 | 0.285 | 3 | 5 |
| 11 | group&share&stock=>acq | 0.99 | 0.285 | 3 | 7 |

TABLE.2: EXAMPLE OF A TRAINING DATA

| id | Document | Class | Random Rank |
|---|---|---|---|
| 1 | real,madrid,stock,loss,share | sport | 1 |
| 2 | real,madrid | sport | 5 |
| 3 | stock,share,group | acq | 2 |
| 4 | stock,share,buy,group | acq | 3 |
| 5 | stock | acq | 4 |
| 6 | iraq,amman,jordan | general | 7 |
| 7 | amman,madrid,real | general | 6 |

In this pruning method (Figure 1), a rule is considered if its antecedent partially matches the training documents words. To describe this pruning method, let's use the above example where the first ranked rule Rule-5: group=>acq is evaluated on the training data shown in Table 2, this rule partially

Input: Given a set of generated rules $R$, and training dataset $T$

Output: classifier ($Cl$)

1 $R'$ = sort($R$);
2 For each rule $r_i$ in $R'$ Do
3 Find all applicable training cases in $T$ that match $r_i$'s condition
4 Insert the rule at the end of Cl
5 Remove all training cases in $T$ covered by $r_i$
6 If $r_i$ cannot correctly cover any training case in $T$
7 Remove $r_i$ from $R$
8 end if
9 end for

matches documents 3 and 4. So it gets inserted into the classifier, and documents 3 and 4 are discarded from the training data set. We proceed to the second ranked rule i.e. Rule-1: real=>sport, we check its applicability with the remaining training documents. We find that Rule-1 partially matches documents 1, 2, and 7. So, it gets inserted into the classifier, and documents 1, 2 and 7 are removed. The third ranked rule Rule-6: amman=>general covers one document 6 so we insert it into the classifier, and we discard document 6 from the training documents set, and repeat the same steps for the rest of the rules until the training documents set becomes empty. In this example, the classifier contains just four significant rules, and the remaining rules will be deleted.

The main difference between the above pruning method and the database coverage [9] is that in the HP method a rule gets inserted into the classifier if it partially covers at least one training case. On the other hand, in the database coverage, a rule must fully match the training case antecedent in order to be inserted.

## IV. Experimental Results

The benchmark used in the experiments is the Reuters-21578 [6]. The Reuters-21578 is the most widely used text data set in the text categorisation research. We used the ModApte version of Reuters-21578. This spilt leads to a corpus of 9,174 documents consisting of 6,603 training and 2,571 testing documents, respectively.

We tested our pruning procedures within the MCAR algorithm on the seven most populated categories with the largest number of documents assigned to them in the training data set. The experiments are conducted on 2.8 Pentium IV machine with 1GB RAM, and the proposed methods and MCAR are implemented using VB.Net programming language with a minsupp and minconf of 2%, and 40%, respectively. The minsupp has been set to 2% since more extensive experiments reported in [9] [8] [16] [15] suggested that it is one of the rates that achieve a good balance between accuracy and the size of the classifiers. The confidence threshold, on the other hand, has a smaller impact on the behaviour of any AC method and it has been set to 40%.

Table 3 represents the number of documents for each category in the Reuters-21578 data set. On these documents we performed stop word elimination but not stemming, and we select the top 1000 features using Chi Square [14]. We performed extensive experiments on the seven most populated categories of the Reuters-21578 text collection to compare HP with the database coverage [9], lazy pruning [3], and the case of no pruning. The bases of the comparison are the number of rules generated and the predictive accuracy.

Table 4 shows the number of rules derived from the Reuters text collection when different pruning approaches are implemented within MCAR algorithm. It is obvious from the numbers shown in Table 4 that in general HP generates less number of rules than lazy approach. However, database coverage derived less number of rules for most of the class labels. One reason behind this is that in database coverage pruning a rule gets inserted into the classifier if its body fully matches one of the training cases. Moreover, the number of rules generated without pruning method on "Acq" class is 80, whereas the number of rules derived using HP pruning procedure is 32. The additional 48 rules produced in the case of no pruning may decrease the classification accuracy and increase the prediction time. It is obvious from the numbers shown in Table 4 that algorithms, which use lazy pruning approach, often generate many more rules than those that employ other approaches. In particular, for all classification data sets we considered, MCAR using lazy pruning produced more rules than other considered pruning heuristics.

One of the principle reasons for generating large number of rules by lazy pruning algorithms is due to

TABLE 3: NUMBER OF DOCUMENTS PER CATEGORY (REUTERS-21578)

| Category Name | Training set | Testing Set |
|---|---|---|
| Acq | 1650 | 719 |
| Crude | 389 | 189 |
| Earn | 2877 | 1087 |
| Grain | 433 | 149 |
| Interest | 347 | 131 |
| Money-FX | 538 | 179 |
| Trade | 369 | 117 |
| **Total** | 6603 | 2571 |

TABLE 4: MCAR NUMBER OF RULES PRODUCED WHEN DIFFERENT PRUNING APPROACHES ARE USED AGAINST REUTER DATA SET

| Class | no Pruning | HP | Database Coverage | Lazy |
|---|---|---|---|---|
| Acq | 80 | 32 | 27 | 40 |
| Crude | 8 | 5 | 4 | 6 |
| Earn | 172 | 29 | 17 | 55 |
| Grain | 5 | 5 | 5 | 5 |
| Interest | 4 | 3 | 2 | 4 |
| Money-FX | 23 | 20 | 12 | 15 |
| Trade | 9 | 8 | 6 | 8 |
| **Total** | **301** | **102** | **73** | **133** |

storing rules that do even cover a single training data case in the classifier. Unlike lazy pruning approach, the database coverage and HP methods eliminate the spare rules and that explains its moderate size classifiers. Specifically, MCAR using our proposed methods and MCAR using database coverage algorithms generate reasonable size classifiers if compared with MCAR using lazy pruning method. This enables domain users to benefit from.

Figure 2 depicts the classification accuracy (%) derived by MCAR algorithm using the different rule pruning methods on the seven most populated categories of Reuters-21578. The accuracy numbers have been generated using a minsupp of 2% and a minconf of 40%. Figure 2 indicates that our proposed HP pruning method outperformed other pruning methods on the given categories. The won-tied-loss records of HP records against no pruning, database coverage and lazy pruning are 7-0-0, 7-0-0 and 7-0-0, respectively. Finally, utilising a partial match pruning approach produces better accuracy than database coverage and lazy pruning.

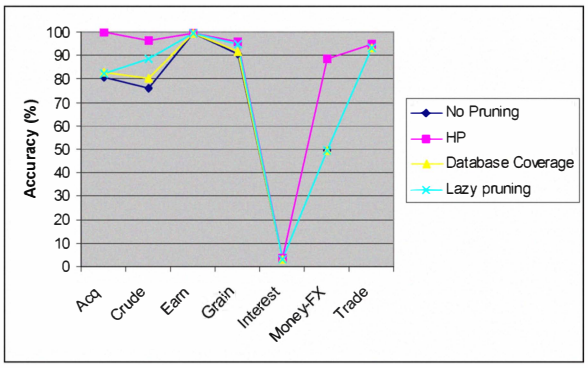


Fig. 2 Accuracy per class label of the Reuter data derived by the MCAR algorithms using the different rule pruning methods

## V. Conclusions.

In this paper, we proposed a new rule pruning method within associative classification mining. We conducted experiments on seven categories selected from the Reuters-21578 text collection using the developed pruning method and other existing methods in associative classification. The bases of the comparison are the number of produced rules and the accuracy, and we implemented all methods within a known associative algorithm called MCAR. The experimental results revealed that the HP outperformed all other pruning techniques with reference to predictive accuracy and number of rules generated. Particularly, HP achieved on average +12.3%, +11.2%, +9.6% higher prediction rates within MCAR algorithm than no pruning, the database coverage and lazy pruning, respectively. In near future, we intend to expand our research to include other rule pruning heuristics in the areas of decision trees, statistics, and rule induction.

## References

[1] Antonie M. and Zaïane O. (2004). An associative classifier based on positive and negative rules. Proceedings of the 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (pp. 64 - 69), Paris, France.

[2] Baralis E., Chiusano S. and Garza P. (2008). A Lazy Approach to Associative Classification. IEEE Trans. Knowl. Data Eng. 20(2): 156-171.

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

[4] Cohen W. (1995). Fast effective rule induction. Proceedings of the 12th International Conference on Machine Learning, (pp. 115-123). CA, USA.

[5] Kundu G., Islam M., Munir S. and Bari M. (2008). ACN: An Associative Classifier with Negative Rules, Computational Science and Engineering, vol. 0, no. 0, (pp. 369-375), 11th IEEE International Conference on Computational Science and Engineering.

[6] Lewis D. (1998). Reuters 21578 text categorisation test collection.http://www.daviddlewis.com/resources/testcollections/reuters21578.

[7] Li B., Li H., Wu M. and Li P. (2008). Multi-label Classification based on Association Rules with Application to Scene Classification, icycs, (pp.36-41), 2008 The 9th International Conference for Young Computer Scientists.

[8] Li W., Han J. and Pei J. (2001). CMAR: Accurate and efficient classification based on multiple-class association rule. Proceedings of the ICDM'01, (pp. 369-376). San Jose, CA.

[9] Liu B., Hsu W. and Ma Y. (1998). Integrating classification and association rule mining. Proceedings of the KDD, (pp. 80-86). New York, NY.

[10] Liu B., Ma Y., Wong, C-K. and Yu. P. (2003). Scoring the data using association rules. Applied Intelligence, 18(2003): 119-135.

[11] Niu Q., Xia S. and Zhang L. (2009). Association Classification Based on Compactness of Rules, wkdd, (pp.245-247), Second International

Workshop on Knowledge Discovery and Data Mining.

[12] Quinlan J. (1998). Data mining tools See5 and C5.0. Technical Report, RuleQuest Research.

[13] Quinlan J. and Cameron-Jones R. (1993). FOIL: A midterm report. Proceedings of the European Conference on Machine Learning, (pp. 3-20), Vienna, Austria.

[14] Snedecor W. and Cochran W. (1989). Statistical Methods, Eighth Edition, Iowa State University Press.

[15] Thabtah, F., Cowling, P., and Peng, Y. (2005) MCAR: Multi-class classification based on association rule approach. Proceeding of the 3rd IEEE International Conference on Computer Systems and Applications (pp. 1-7).Cairo, Egypt.

[16] Thabtah, F., Cowling, P., and Peng, Y. (2004) MMAC: A new multi-class, multi-label associative classification approach. Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM '04), (pp. 217-224). Brighton, UK. (Nominated for the Best paper award).

[17] Xu X., Han G. and Min H. (2004). A novel algorithm for associative classification of images blocks. Proceedings of the fourth IEEE International Conference on Computer and Information Technology, (pp. 46-51). Lian, Shiguo, China.

[18] Yin X. and Han J. (2003). CPAR: Classification based on predictive association rule. Proceedings of the SDM (pp. 369-376). San Francisco, CA.