Generic Associative Classification Rules: A Comparative Study

I. Bouzouita and S. Elloumi

Faculty of Sciences of Tunis,
Computer Science Department, 1060 Tunis, Tunisia.
ines.bouzouita@fst.rnu.tn, samir.elloumi@fst.rnu.tn

Abstract

Associative classification is a supervised classification approach, integrating association mining and classification. Several studies in data mining have shown that associative classification achieves higher classification accuracy than do traditional classification techniques. However, the associative classification suffers from a major drawback: The huge number of the generated classification rules which takes efforts to select the best ones in order to construct the classifier. To overcome such drawback, we have proposed an associative classification method that reduces associative classification rules without jeopardizing the classification accuracy. Moreover, we will introduce in this paper two different strategies to classify new instances based on some interestingness measures that arise from data mining literature in order to select the best rules during classification. A detailed description of this method is presented in this paper, as well as the experimentation study on 12 benchmark data sets proving that our approach is highly competitive in terms of accuracy in comparison with popular classification approaches.

Keywords: *Associative Classification, Classification Rules, Generic association rules, Classifier, interestingness measures, robustness.*

1. Introduction

Classification is an activity that Human beings normally excel in. In fact, everybody do it almost all the time, and without conscious effort. We receive information via our various sensory organs, which is processed instantaneously by our brain so that, almost immediately, we are able to classify the data, without having made any perceptible effort. What is even more impressive is the accuracy with which we can perform classification tasks even under non-ideal conditions, for instance, when the information that needs to be processed is vague, imprecise or even incomplete. In fact, most of our day-to-day activities are based on our success in performing various classification tasks.

Indeed, our brain simplifies our perceptions of the environment or the reflections by grouping resemblances objects or concepts. Should the opposite occur, we would become totally ineffective because the huge quantity of information that we handle every second. That is why it is indispensable to select and to organize into a hierarchy to be able to make a decision.

For example if we consider the impact of any action as active on my environment, we can classify the activities according to the consequences which they entail.

The discipline of supervised classification in machine learning area deals with the problem of developing algorithms and approaches that can en- able the computer-implementation of many of the classification tasks that humans normally perform. The motivation is to perform these tasks more accurately, or faster, and perhaps more economically than humans and, in many cases, to release them from drudgery resulting from performing routine classification tasks repetitively and mechanically. The goal of classification research is to devise ways and means of automating certain decision-making processes that lead to classification and prediction.

Supervised classification involves constructing a model from classified objects, in order to classify previously unseen objects as accurately as possible.

Associative classification is a supervised classification method integrating association mining and classification. Many experimental studies have shown that associative classification is a promising approach. However, the latter suffer from a major drawback: the huge number of the generated classification rules which takes efforts to select the best ones in order to construct the classifier.

Given a training data set, the task of an associative classification algorithm is to discover the classification rules which satisfy the user specified constraints denoted respectively by minimum support (*minsup*) and minimum confidence (*minconf*) thresholds. The classifier is built by choosing a subset of the generated classification rules that could be of use to classify new objects or instances. Many studies have shown that AC often achieves better accuracy than do traditional classification techniques [15, 14]. In fact, it could discover interesting rules omitted by well known approaches such as C4.5 [12]. However, the main drawback of this approach is that the number of generated associative classification rules could be large and takes efforts to retrieve, prune, sort and select high quality rules among them. To overcome this problem, we propose a new approach which extracts generic classification rules in order to retain a small set of rules with higher quality and lower redundancy in comparison with current AC approaches. The main originality of this is that it extracts the generic classification rules directly from a training set with- out using generic basis of association rules. Furthermore, we propose a robustness study of classifier.

This tackled issue is quite challenging, since the goal is to use generic rules while maintaining high classifier accuracy.

The remainder of the paper is organized as follows. Section 2 briefly reports basic concepts of associative classification, scrutinizes related pioneering works by splitting them into two groups.

Section 3 presents our proposed approach, where details about classification rules discovery and two different building classifiers are proposed. Experimental results and comparisons are given in section 4. Finally, section 5 concludes this paper and points out future perspectives.

2. Basic Notions and Related Work

2.1 Basic Notions

Let us define the classification problem in an association rule task. Let *D* be the training set with n attributes (columns) $A_1, .., A_n$ and *D* rows.

Let *C* be the list of class attributes.

Formal context: A formal context is a triplet = (, ,), where . represents a finite set of transactions, is a finite set of items and is a binary (incidence) relation (*i.e.*, . $\subseteq$ . .). Each couple $(o, i) \in$. expresses that the transaction $o \in$ contains the item $i \in$. Within the context shown by Table 1, objects are denoted by numbers and attributes by letters.

We define two functions summarizing links between subsets of objects and subsets of attributes induced by , that maps sets of objects to sets of attributes and *vice versa*. Thus, for a set $O \in$, we define $\phi(O) =$ $i \in I$ $\forall o, o \in . \Rightarrow (o, i) \in$ and for $i \in$, $\psi(I) =$ $o \in$ $\forall i,$ $i \in . \Rightarrow (o, i) \in$.

Both operators $\phi(O)$ and $\psi(I)$, form a *Galois connection* between the sets ( ) and ( ) [6]. Consequently, both compound operators ϕ and ψ are closure operators in particular $\omega = \phi \circ \psi$ is a closure operator.

Frequent closed itemset: An itemset $I \subseteq$ is said to be *closed* if $\omega(I) = I$. I is said to be *frequent* if its *relative support*, Support(I) = $\underline{\psi(I)}$, exceeds a user-defined minimum threshold, denoted *minsup*.

Minimal generator [4]: An itemset $g \subseteq$ is said to be *minimal generator* of a closed itemset f , if and only if $\omega(g) = f$ and there is no $g_1 \subseteq g$ such that $\omega(g_1) = f$. The set f of the minimal generators of f is: $f =$ $g \subseteq .$ $\omega(g) = f \wedge g_1 \subset g$ such as $\omega(g_1) = f$.

An association rule is a relation between itemsets having the following form: $R : X \Rightarrow Y - X$, where X and Y are frequent itemsets for a minimal support *minsup*, and $X \subset Y$. Itemsets X and $(Y–X)$ are called, respectively, *premise* and *conclusion* of the rule R. An association rule is valid whenever its strength metric, confidence(R)=$\underline{support(Y)}/\ support(X)$, is greater than or equal to the minimal threshold of confidence *minconf*.

An associative classification rule (ACR) is a special case of an association rule. In fact, an ACR conclusion part is reduced to a single item referring a class attribute. For example, in an ACR such as $X \Rightarrow c_i$, c_i must be a class attribute.

Definition 1. *An object or instance in D can be described as a combination of attribute names and values a_i and an attribute class denoted by c_i [10].*

Definition 2. *An item is described as an attribute name and a value a_i [10].*

Definition 3. *An itemset can be described as a set of items contained in an object.*

Definition 4. *An associative classification rule is of the form: $A_1, A_2, ..., A_n \Rightarrow c_i$ where the premise of the rule is an itemset and the conclusion is a class attribute.*

A classifier is a set of rules of the form $A_1, A_2, ..., A_n \Rightarrow c_i$ where A_i is an attribute and c_i is a class attribute. The classifier should be able to predict, as accurately as possible, the class of an unseen object belonging to the test data set. In fact, it should maximise the equality between the predicted class and the hidden actual class.

The AC achieves higher classification accuracy than do traditional classification approaches [15, 14]. The classification model is a set of rules easily understandable by humans and that can be edited [15, 14].

2.2 Related Work

Associative classification approaches can be categorized into two groups according to the way of the classification rules extraction:

1. **Two stages algorithms:** Algorithm 1 indicates the different steps of an associative classification two stages algorithm. In the first stage, a set of associative

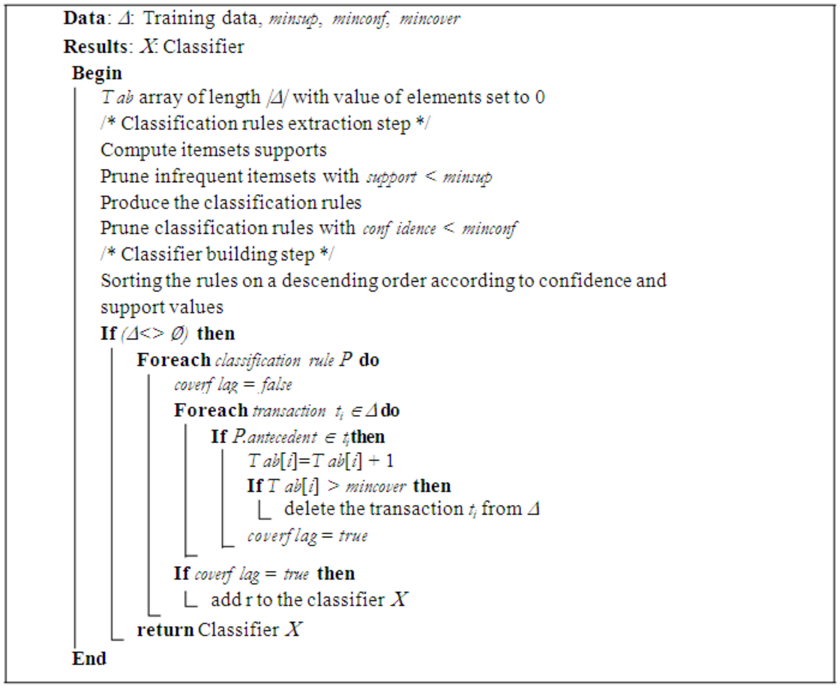
classification rules is produced. Then, this latter is pruned and placed into a classifier. Examples of such approaches are CBA [10], CMAR [9], ARC-AC and ARC-BC [3, 2].

CBA [10] was one of the first algorithms to use association rule approach for classification. This approach, firstly, generates all the association rules with certain support and confidence thresholds as candidate rules by implementing the Apriori algorithm [1]. Then, it selects a small set from them by evaluating all the generated rules against the training data set. When predicting the class attribute for an example, the highest confidence rule, whose body is satisfied by the example, is chosen for prediction.

CMAR [9] generates rules in a similar way as CBA with the exception that CMAR introduces a CR-tree structure to handle the set of generated rules and uses a set of them to make a prediction using a weighted $\chi 2$ metric [9]. The latter metric evaluates the correlation between the rules.

ARC-AC and ARC-BC have been introduced in [3, 2] in the aim of text categorization. They generate rules similar to the Apriori algorithm and rank them in the same way as do CBA rules ranking method. ARC-AC and ARC-BC calculate the average confidence of each set of rules grouped by class attribute in the conclusion part and select the class attribute of the group with the highest confidence average.

GARC [5] extracts the generic basis of association rules. Once obtained, generic rules are filtered out to retain only rules whose conclusions include a class attribute. Then, by applying the decomposition axiom, we obtain generic classification rules. After that, a total order is set on them. The data set coverage is similar to that in CBA. In fact, a data object of the training set is removed after it is covered by a selected generic rule.



```
Data: $\Delta$: Training data, $minsup$, $minconf$, $mincover$
Results: $X$: Classifier
Begin
    $T ab$ array of length /$\Delta$/ with value of elements set to 0
    /* Classification rules extraction step */
    Compute itemsets supports
    Prune infrequent itemsets with $support < minsup$
    Produce the classification rules
    Prune classification rules with $conf idence < minconf$
    /* Classifier building step */
    Sorting the rules on a descending order according to confidence and
    support values
    If $(\Delta <> \emptyset)$ then
        Foreach $classification\ rule\ P$ do
            $coverf\ lag = false$
            Foreach $transaction\ t_i \in \Delta$ do
                If $P.antecedent \in t_i$ then
                    $T ab[i] = T ab[i] + 1$
                    If $T ab[i] > mincover$ then
                        delete the transaction $t_i$ from $\Delta$
                    $coverf lag = true$
            If $coverf\ lag = true$ then
                add r to the classifier $X$
        return Classifier $X$
End
```

Algorithm 1: Generic associative classification two stages algorithm

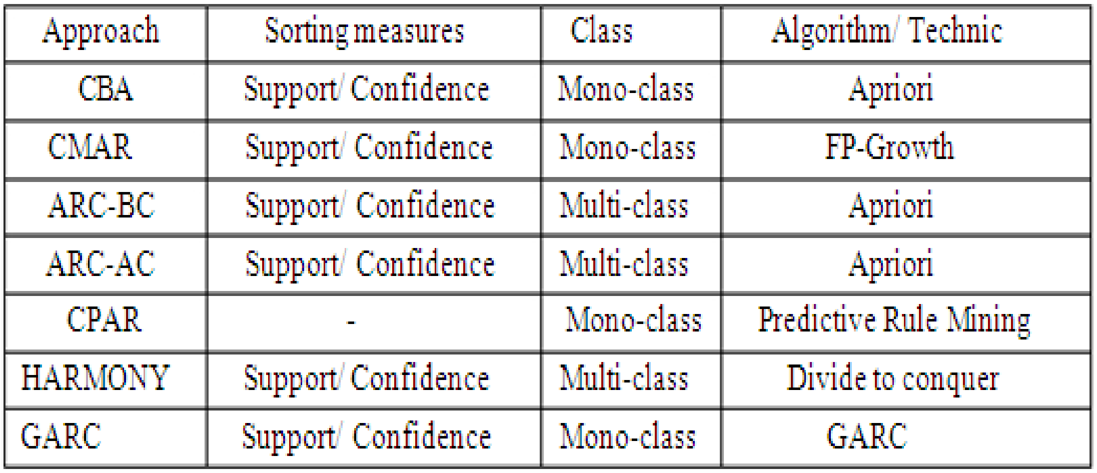
2. **Integrated algorithms:** The classifier is produced in a single processing step. Examples of such approaches are CPAR [14] and Har- mony [13].

   The CPAR [14] algorithm adopts FOIL [11] strategy in generating rules from data sets. It seeks for the best rule itemset that brings the highest gain value among the available ones in data set. Once the itemset is identified, the examples satisfying it will be deleted until all the examples of the data set are covered. The searching process for the best rule itemset is a time consuming process, since the gain for every possible item needs to be calculated in order to determine the best item gain. During rule generation step, CPAR derives not only the best itemset but all close similar ones. It has been claimed that CPAR improves the classification accuracy whenever compared to popular associative methods like CBA and CMAR [14].

   An AC approach called Harmony was proposed in [13]. Harmony uses an instance-centric rule generation to discover the highest confidence discovering rules. Then, Harmony groups the set of rules into k groups according to their rule conclusions, where k is the total number of distinct class attributes in the training set. Within the same group of rules, Harmony sorts the rules in the same order as do CBA. To classify a new test instance, Harmony computes a score for each group of rules and assigns the class attribute with the highest score or a set of class attributes if the underlying classification is a multi-class problem. It has been claimed that Harmony improves the effciency of the rule generation process and the classification accuracy if compared to CPAR [14].

In the remainder, we recapitulate the surveyed approaches by addressing Table 2.

Table 2. Features of the Surveyed Associative Classification Approaches



Approach	Sorting measures	Class	Algorithm/ Technic
CBA	Support/ Confidence	Mono-class	Apriori
CMAR	Support/ Confidence	Mono-class	FP-Growth
ARC-BC	Support/ Confidence	Multi-class	Apriori
ARC-AC	Support/ Confidence	Multi-class	Apriori
CPAR	-	Mono-class	Predictive Rule Mining
HARMONY	Support/ Confidence	Multi-class	Divide to conquer
GARC	Support/ Confidence	Mono-class	GARC

It is noteworthy that all the approaches, except CPAR, sort the classification rules using support and confidence measures in order to build the classifier. However, the support confidence measures could be mislead- ing while classifying new objects. Moreover, all the approaches, except GARC, manipulate the totality number of the classification rules. Thus, regardless of the methodology used to generate the classifier, there are an overwhelming number of rules manipulated during the learning stage, which is the main problem with AC approaches. In order to overcome this drawback, our proposed approach deduced from GARC approach, tries to gouge this fact by generating a classifier composed from generic

(ACR) directly from a training set. In the following, we will introduce our approach after recalling some key notions about the Formal Concept Analysis (FCA), a mathematical tool necessary for the comprehension of the two versions of our proposed approach.

3. Our Proposed Approach Based on Generic Associative Classification Rules

In this section, we propose two versions of our proposed AC method. The intuition behind this approach is that we can extract the classifier directly from a training set. Our approach permits:

- to apply the cover principle while generating (ACR). Suppose a training object is covered by the generated rule so it will be removed from the training set. This permits to stop the generic (ACR) generation process if all the training objects are removed.
- to treat only frequent closed itemsets which include a class label in order to generate generic (ACR) and thus to avoid the extraction of the whole set of generic association rules.

In the following, we will introduce our approach after a brief description of FCA mathematical background necessary for the comprehension of this work.

3.1 Basic Definitions

Interested reader for key results from the Galois lattice-based paradigm in FCA is referred to [6].

In the following, we will present and explain in details the our approach presented by algorithm 2.

3.2 Learning Stage

Given a training data set shown by Figure 3 **(a)**, a minimum support (*minsup*=$\underline{1}$) and a minimum confidence (*minconf* =$\underline{2/3}$), our approach 3 generates the associated Iceberg Galois Lattice depicted by Figure 2.

The Iceberg Galois Lattice represents precedence-relation-based ordered closed itemsets. Each closed itemset is decorated with its associated list of minimal generators. Each node in the Iceberg containing a class label will be treated to generate generic association rules. Then, we decompose the rule in order to obtain generic classification rule of the form $a_i, a_j, ..., a_n \Rightarrow c_i$. Hereafter, we present two versions of our approach to extract generic associative classification rules from Iceberg Galois Lattice.

Let be the set of frequent closed itemsets including a class label and f be the set of minimal generators of all the frequent itemsets included or equal to a closed frequent itemset including a class label f.

Version1: Our approach-version1 = R: $g_S \Rightarrow (f_1 - g_S)$ $f, f_1 \in$.
and $(f - g_S) = \emptyset$ and $g_S \in$ f and f_1 f and confidence(R)$\geq$*minconf*
.

Version2: Our approach-version2 = R: $g_S \Rightarrow (f_1 - g_S)$ $f, f_1 \in$.
and $(f - g_S) = \emptyset$ and $g_S \in$ f and f_1 f and confidence(R)$\geq$*minconf*

and $g \subset g_S$ such that confidence$(g \Rightarrow f_1\text{-}g) \geq minconf$.

Our approach applies the cover principle by searching and removing training objects whose bodies include the rule premise.While, it exists training objects the process is continuing. Our approach-version2 is more compact then Our approach-version1 and gives rules with mini- mal premise since the latter is represented by one of the smallest frequent minimal generators satisfying *minconf* threshold. Once the generic clas- sification rules are obtained, a total order on rules is set as follows. Given two rules R_1 and R_2, R_1 is said to precede R_2, denoted $R_1 > R_2$ if the followed condition is fulfilled:

- $conf\ idence(R_1) > confidence(R_2)$ or
- $conf\ idence(R_1) = confidence(R_2)$ and $support(R_1) > support(R_2)$ or
- $conf\ idence(R_1) = confidence(R_2)$ and $support(R_1) = support(R_2)$ and R_1 is generated before R_2.

The resulting classifier is given by Figure 3 **(b)**.

3.3 Classifying Stage

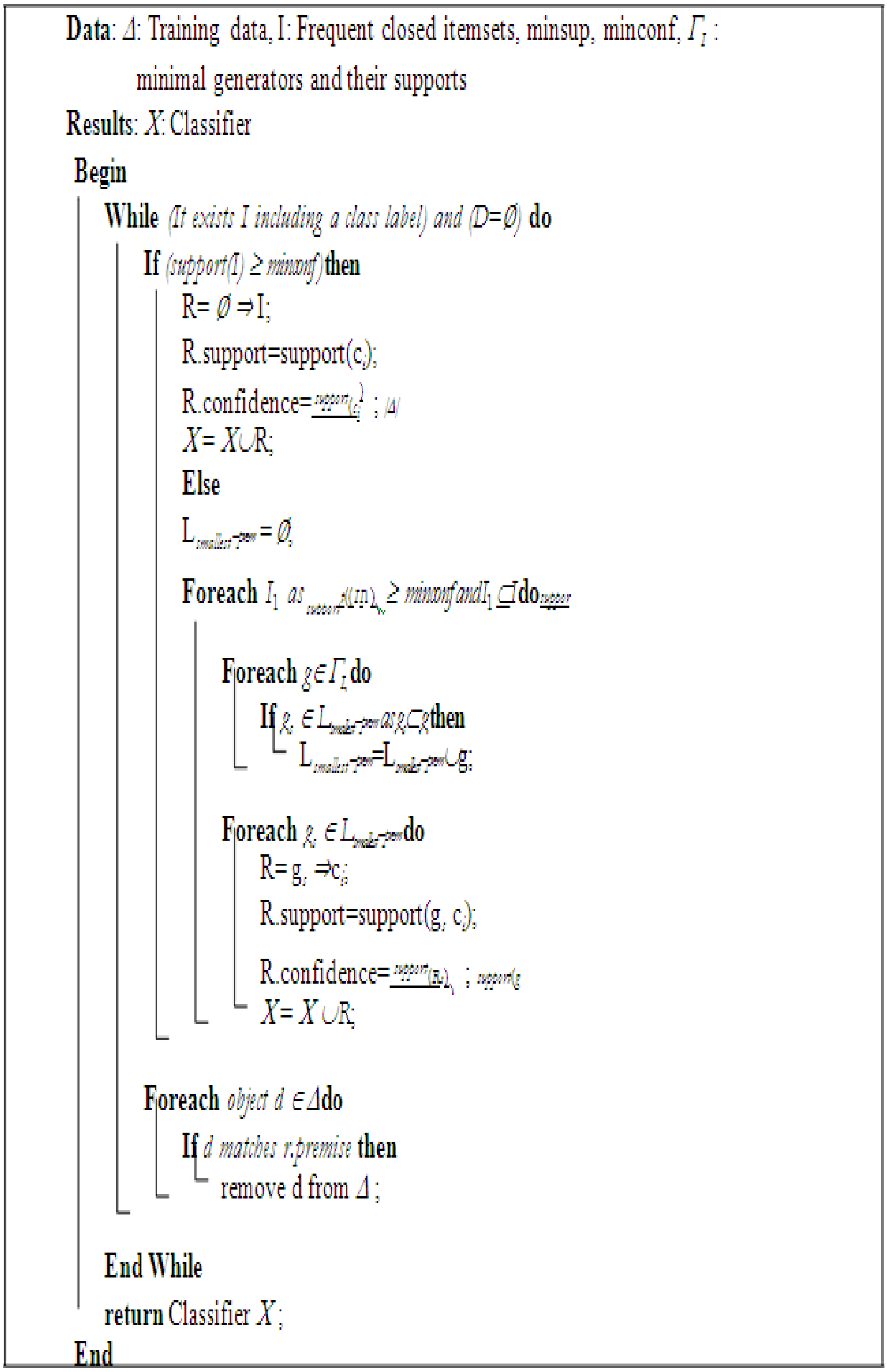
After a set of rules is selected for classification, our algorithm is ready to classify new objects. Some methods such as those described in [10, 3, 2, 13] are based on the support-confidence order to classify a new object. However, the confidence measure selection could be misleading, since it may identify a rule A ⇒ B as an interesting one even though, the occurrence of A does not imply the occurrence of B [7]. In fact, the confidence can be deceiving since it is only an estimate of the conditional probability of itemset B given an itemset A and does not measure the actual strength of the implication between A and B.

We notice that better results are obtained by our proposed algorithm when compared to Garc algorithm thanks to the integration of the cover principle in the learning stage and the reduction of the treatment to the frequent closed itemsets including a class label when generating generic (ACR).

Let us consider the following parameters:

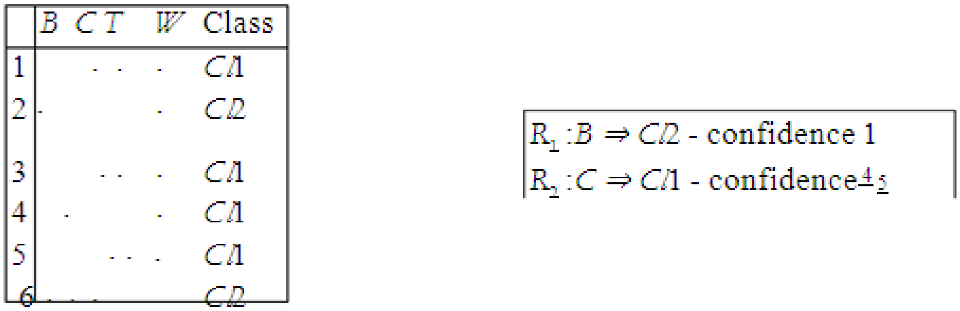
- n: number of frequent closed itemsets;
- no: number of training objects;
- t: cardinality average of a frequent closed itemset;
- l: cardinality average of smallest premise list;
- ng: average of minimal generators number associated to a frequent closed itemset.

After simplification, complexity of our algorithm is equal to $O(min(\text{no}, \text{n}) \, . \, \text{n} \quad \text{t}^2 \quad \text{ng} \quad \text{l})$.



```
Data: $\Delta$: Training data, I: Frequent closed itemsets, minsup, minconf, $\Gamma_I$ :
        minimal generators and their supports
Results: $X$: Classifier
Begin
    While (It exists I including a class label) and ($D=\emptyset$) do
        If (support(I) ≥ minconf) then
            R= $\emptyset \Rightarrow$ I;
            R.support=support(c_i);
            R.confidence= $\frac{support(c_i)}{}$ ; $|\Delta|$
            $X = X \cup R$;
            Else
            $L_{smallest-gen} = \emptyset$;
            Foreach $I_1$ as $_{support}$ (I1) ≥ minconf and $I_1 \sqsupset I$ do $_{support}$
                Foreach $g \in \Gamma_I$ do
                    If $g_i \in L_{smallest-gen}$ as $g_i \subset g$ then
                        $L_{smallest-gen} = L_{smallest-gen} \cup g$;
                Foreach $g_i \in L_{smallest-gen}$ do
                    R= $g_i \Rightarrow c_i$;
                    R.support=support($g_i$, $c_i$);
                    R.confidence= $\frac{support(R)}{}$ ; support(g
                    $X = X \cup R$;
        Foreach object $d \in \Delta$ do
            If d matches r.premise then
                remove d from $\Delta$ ;
    End While
    return Classifier $X$;
End
```

Algorithm 2: Our proposed algorithm



	B	C	T	W	Class
1		·	·	·	$Cl1$
2	·			·	$Cl2$
3		·	·	·	$Cl1$
4		·		·	$Cl1$
5		·	·	·	$Cl1$
6	·	·	·		$Cl2$

$R_1 : B \Rightarrow Cl2$ - confidence 1
$R_2 : C \Rightarrow Cl1$ - confidence $\frac{4}{5}$

Fig. 1: (a) Training data. (b) Our proposed Classifier

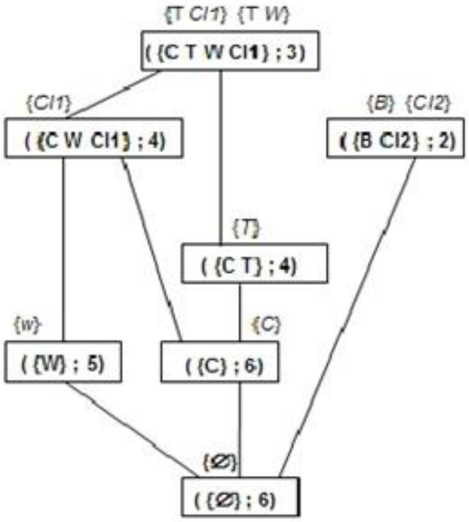
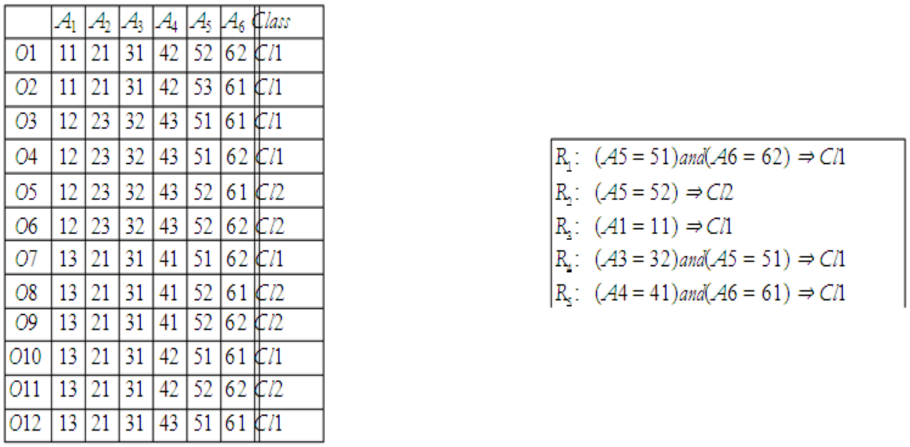


Fig. 2. Iceberg Galois Lattice

Example 1. The training data set shown by figure 3 (Left) is composed of twelve objects. Each object is described by six categorical attributes and belongs to a class. For example, the object $O1$ has the attributes $A1$ = 11, $A2$ = 21, $A3$ = 31, $A4$ = 42, $A5$ = 52, $A6$ = 62 and belongs to the class $Cl1$. We have set the *minsup* and *minconf* values to 1% and 80%, respectively. We extract generic classification rules by apply- ing version2 definition. Then, we generate generic ACR by applying the decomposition axiom to obtain rules of the form $A_1, A_2, ..., A_6 \Rightarrow c_i$ with $c_i \in$ $Cl1, Cl2$. Once the generic ACR obtained, we set a descending order on them according to the support and confidence values. Then, we apply the cover algorithm where each object of the training data . had to be satisfied (covered) by one rule before it is no longer considered in the classifier generation process and removed from the training data. The resulting classifier is given by figure 3 (Right).



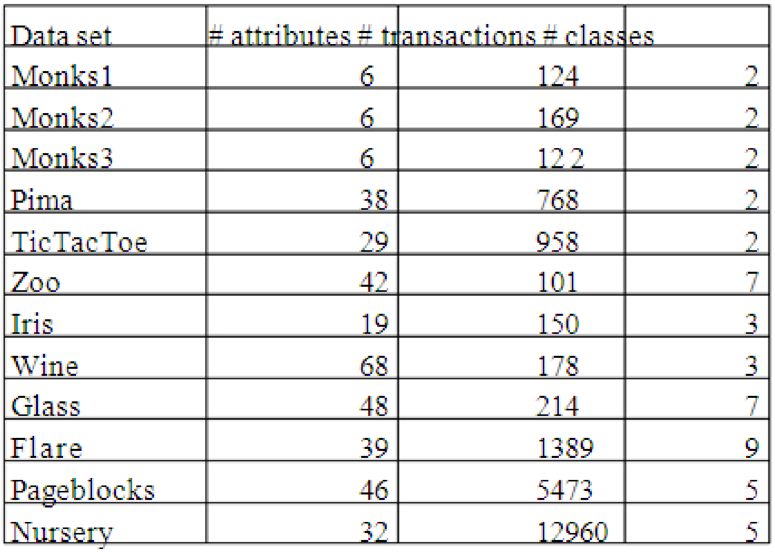
	A_1	A_2	A_3	A_4	A_5	A_6	Class
O1	11	21	31	42	52	62	Cl1
O2	11	21	31	42	53	61	Cl1
O3	12	23	32	43	51	61	Cl1
O4	12	23	32	43	51	62	Cl1
O5	12	23	32	43	52	61	Cl2
O6	12	23	32	43	52	62	Cl2
O7	13	21	31	41	51	62	Cl1
O8	13	21	31	41	52	61	Cl2
O9	13	21	31	41	52	62	Cl2
O10	13	21	31	42	51	61	Cl1
O11	13	21	31	42	52	62	Cl2
O12	13	21	31	43	51	61	Cl1

R_1: $(A5 = 51) and (A6 = 62) \Rightarrow Cl1$
R_2: $(A5 = 52) \Rightarrow Cl2$
R_3: $(A1 = 11) \Rightarrow Cl1$
R_4: $(A3 = 32) and (A5 = 51) \Rightarrow Cl1$
R_5: $(A4 = 41) and (A6 = 61) \Rightarrow Cl1$

Fig. 3 Left: Training data. Right: our proposed classifier for minsup=1% and minconf =80%.

4. Experimental Study

We have conducted experiments to evaluate the accuracy of our pro- posed approach. Experiments were conducted using 12 data sets taken from UCI Machine Learning Repository(1). The chosen data sets were discretized using the LUCS-KDD (2) software. All the ex- periments were performed on a 2.4 GHz Pentium IV PC under Redhat Linux 7.2.



Data set	# attributes	# transactions	# classes
Monks1	6	124	2
Monks2	6	169	2
Monks3	6	12 2	2
Pima	38	768	2
TicTacToe	29	958	2
Zoo	42	101	7
Iris	19	150	3
Wine	68	178	3
Glass	48	214	7
Flare	39	1389	9
Pageblocks	46	5473	5
Nursery	32	12960	5

Table 3. Data Set Description

Classification accuracy can be used to evaluate the performance of classification methods. It is the percentage of correctly classified examples in the test set and can be measured by splitting the data sets into a training set and a test set.

During experiments, we have used available test sets for data sets Monks1, Monks2 and Monks3 and we applied the 10 cross-validation for the rest of data sets, in which a data set is divided into 10 subsets; each sub- set is in turn used as testing data while the remaining data is used as the training data set; then the average accuracy across all 10 trials is reported.

During these experiments, we evaluated the percentage of accuracy *vs* both *minsup* and *minconf* values variation. We compared the effectiveness of the two versions for the classification framework. For this, we conducted experiments with reference to accuracy.

4.1 Variation of Interestingness Measures Thresholds

Table 9 sketches the variation of the accuracy *vs* both *minsup* and *minconf*. From the reported statistics, we remark that in most cases by increasing the minsup value, the accuracy value diminishes. This is explained by the fact that ACR with low support value are useful to correctly classify new objects. That's why; it is interesting to decrease the minsup value when generating ACR. However, by decreasing the *minsup* value, the number of ACR increases, which is the main problem of AC approaches.

In fact, our approach uses generic ACR which allow to reduce the number of the manipulated rules in comparison with earlier AC approaches even with low *minsup* value. Thus, we avoid the pruning of rules with low support value since they can classify correctly new objects.

Table 5 represents a comparison between version1 and version2.

The best average accuracy, highlighted in bold print, is given by version2.

The main reason for this is that version2 classifier contains generic rules with small premises which are beneficial for classification process.

Furthermore, as shown in Table 6, the number of rules generated by version2 is less than that generated by version1. In the following, we put the focus on comparing our proposed approach accuracy versus that of the well known classifiers ID3, C4.5, CBA and Harmony.

4.2 Generic Classification Rules Impact

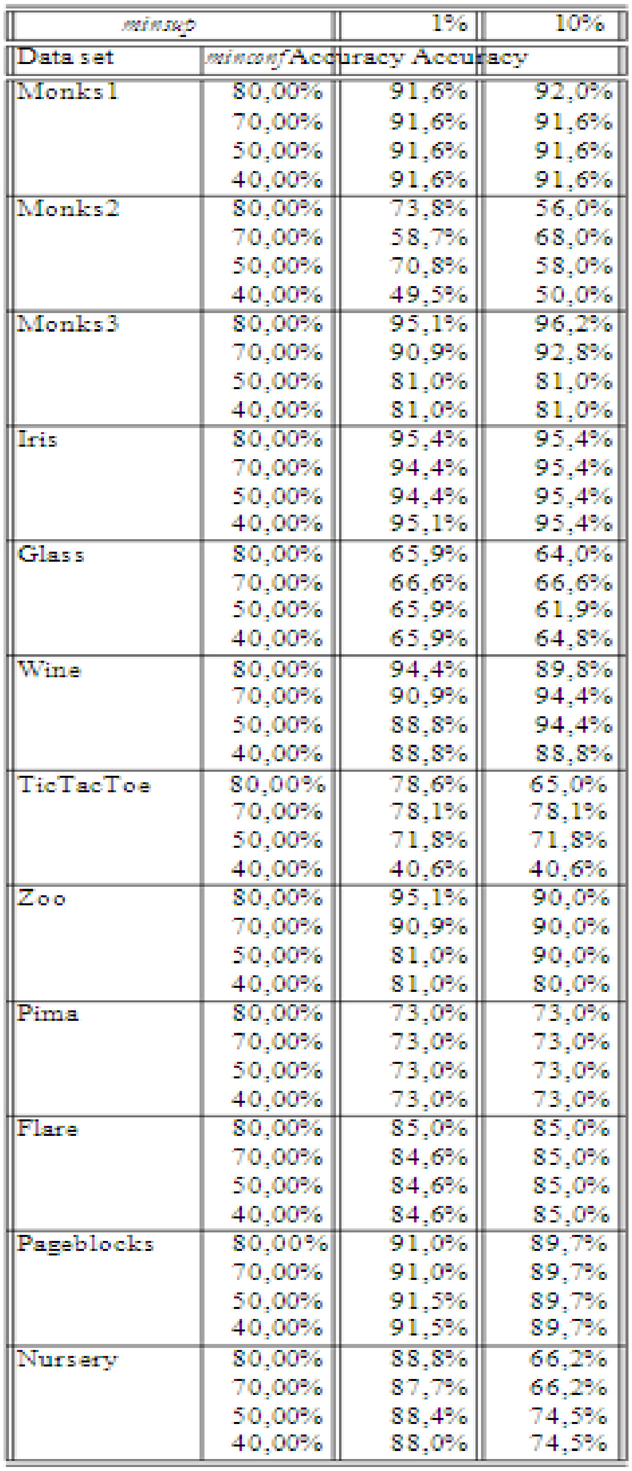
Table 7 represents the accuracy of the classification systems generated by ID3, C4.5, CBA, Harmony and our approach on the twelve benchmark data sets. The best accuracy values obtained for each of data sets is highlighted in bold print. Table 7 shows that our approach outperforms the traditional classification approaches, *i.e.*, ID3 and C4.5 on six data sets and the associative classification approaches on nine data sets.

Statistics depicted by Table 7 confirm the fruitful impact of the use of the generic rules. The main reason for this is that the classifier based on our approach contains generic rules with small premises. In fact, this kind of rule allows to classify more objects than those with large premises.

[1] *Available at* http://www.ics.uci.edu/~mlearn/MLRepository.html

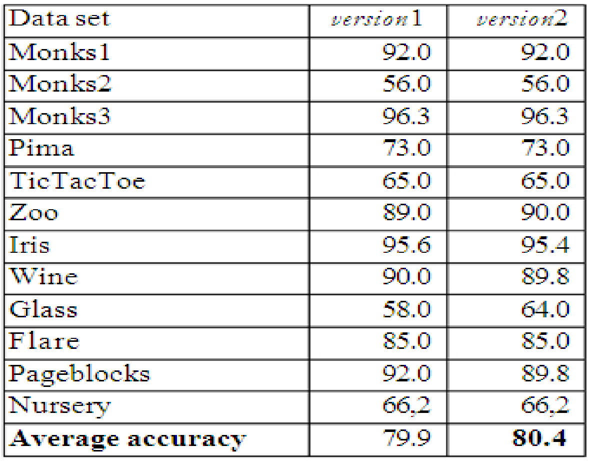
[2] *Available at* http://www.csc.liv.ac.uk/~frans/KDD/Software/LUCS-KDD-DN/lucs-kdd DN.html

Table 4. Evaluation of the Percentage of Accuracy vs both *minsup* and *minconf* Values Variation



minsup		1%	10%
Data set	*minconf*	Accuracy	Accuracy
Monks1	80,00%	91,6%	92,0%
	70,00%	91,6%	91,6%
	50,00%	91,6%	91,6%
	40,00%	91,6%	91,6%
Monks2	80,00%	73,8%	56,0%
	70,00%	58,7%	68,0%
	50,00%	70,8%	58,0%
	40,00%	49,5%	50,0%
Monks3	80,00%	95,1%	96,2%
	70,00%	90,9%	92,8%
	50,00%	81,0%	81,0%
	40,00%	81,0%	81,0%
Iris	80,00%	95,4%	95,4%
	70,00%	94,4%	95,4%
	50,00%	94,4%	95,4%
	40,00%	95,1%	95,4%
Glass	80,00%	65,9%	64,0%
	70,00%	66,6%	66,6%
	50,00%	65,9%	61,9%
	40,00%	65,9%	64,8%
Wine	80,00%	94,4%	89,8%
	70,00%	90,9%	94,4%
	50,00%	88,8%	94,4%
	40,00%	88,8%	88,8%
TicTacToe	80,00%	78,6%	65,0%
	70,00%	78,1%	78,1%
	50,00%	71,8%	71,8%
	40,00%	40,6%	40,6%
Zoo	80,00%	95,1%	90,0%
	70,00%	90,9%	90,0%
	50,00%	81,0%	90,0%
	40,00%	81,0%	80,0%
Pima	80,00%	73,0%	73,0%
	70,00%	73,0%	73,0%
	50,00%	73,0%	73,0%
	40,00%	73,0%	73,0%
Flare	80,00%	85,0%	85,0%
	70,00%	84,6%	85,0%
	50,00%	84,6%	85,0%
	40,00%	84,6%	85,0%
Pageblocks	80,00%	91,0%	89,7%
	70,00%	91,0%	89,7%
	50,00%	91,5%	89,7%
	40,00%	91,5%	89,7%
Nursery	80,00%	88,8%	66,2%
	70,00%	87,7%	66,2%
	50,00%	88,4%	74,5%
	40,00%	88,0%	74,5%

Table 5. Accuracy Comparison of Version 1 and Version 2 of our Proposed Algorithms for *minsup*=10% and *minconf* =80%



Data set	*version*1	*version*2
Monks1	92.0	92.0
Monks2	56.0	56.0
Monks3	96.3	96.3
Pima	73.0	73.0
TicTacToe	65.0	65.0
Zoo	89.0	90.0
Iris	95.6	95.4
Wine	90.0	89.8
Glass	58.0	64.0
Flare	85.0	85.0
Pageblocks	92.0	89.8
Nursery	66,2	66,2
Average accuracy	79.9	**80.4**

4.3 Robustness of Our Proposed Approach

In order to get more realistic data sets, we introduced noise in the afore- mentioned databases, and this by reversing the class attribute of succes- sively 10%, 20%, and 30% of each data set objects. Table9 sketches results obtained by varying the correlation measures which allows choosing the best rule to classify a new object in noisy data. The used measures are the following ones:

- The lift metric [7] computes the correlation between A and B as follows:

$$\mathit{lift}(A \Rightarrow B) = \mathit{support}(A) * \mathit{support}(B) = 0.250.\underline{201}.900 = 0.893\ \underline{\mathit{support}(AB)\ }*0$$

The fact that this quantity is less than 1 indicates negative correlation between A and B.

If the resulting value is greater than 1, then A and B are said pos- itively correlated. If the resulting value is equal to 1, then A and B are independent and there is no correlation between them.

- The least confidence (or surprise) [8] metric is computed as follows:

Surprise ($A \Rightarrow B$) = (support (AB) - support (A *B*))/ support (B)

logical rule: surprise ($A \Rightarrow B$) = P (A)/ P (B)

A and B independent: surprise ($A \Rightarrow B$) = 2 P (A) - (P (A)/ P (B))

A and B incompatible: surprise ($A \Rightarrow B$) = - P (A)/ P (B)

Surprise metric selects rules, even with small support value, having the premise A always with the conclusion B and nowhere else.

- Loevinger metric [8] is computed as follows:

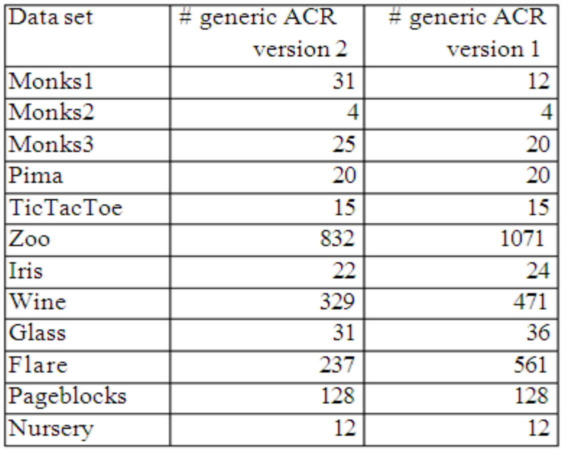
loevinger($A \Rightarrow B$) = (P(B/A) -P(B))/ P(*B*)

Unlike confidence metric, Loevinger metric does not sufer form the problem of producing misleading rules.

Table 9 shows that lift measure is the most stable among the studied measures. This is explained by the fact that the lift of a rule is a relative measure in the sense that it compares the degree of dependence in a rule versus independence between the consequent items and the antecedent items. The rules that have higher lift will have higher dependence in them.

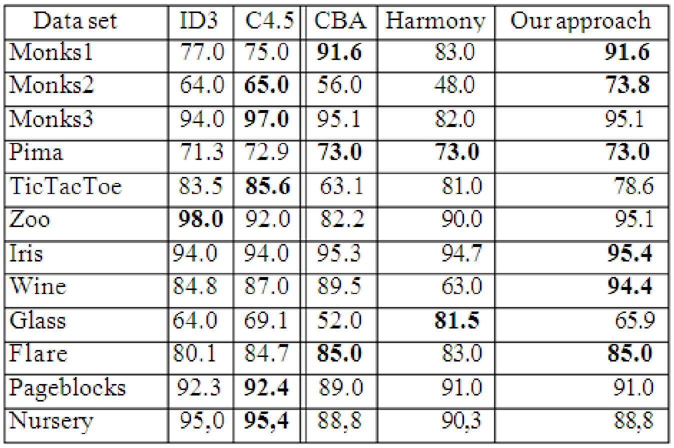
Robustness study results are shown by Figure 4. Graphics of this Figure illustrate variation of accuracy vs noisy data. Our proposed associative classification approach confirms the stability of accuracy of our classifier in spite of the presence of noisy data. Indeed, the vertical axis gives accuracy percentage and the horizontal one gives noisy data percentage. The graph at the top concerns our approach and shows that this last one presents a more stable accuracy and particularly for bases monks2 and monks 3. On the other hand the approaches c4.5 and id3 show a greater decline of accuracy for the same percentage of noise. This is explained by the usefulness of the right choice of the appropriate interestingness measures to evaluate the quality of classification rules of our classifier.

Table 6. Number of associative classification rules for *minsup*=10% and *minconf* =80%



Data set	# generic ACR version 2	# generic ACR version 1
Monks1	31	12
Monks2	4	4
Monks3	25	20
Pima	20	20
TicTacToe	15	15
Zoo	832	1071
Iris	22	24
Wine	329	471
Glass	31	36
Flare	237	561
Pageblocks	128	128
Nursery	12	12

Table 7. Accuracy comparison of ID3, C4.5, CBA, Harmony and our approach algorithms for *minsup*=1% and *minconf* =80%



Data set	ID3	C4.5	CBA	Harmony	Our approach
Monks1	77.0	75.0	**91.6**	83.0	**91.6**
Monks2	64.0	**65.0**	56.0	48.0	**73.8**
Monks3	94.0	**97.0**	95.1	82.0	95.1
Pima	71.3	72.9	**73.0**	**73.0**	**73.0**
TicTacToe	83.5	**85.6**	63.1	81.0	78.6
Zoo	**98.0**	92.0	82.2	90.0	95.1
Iris	94.0	94.0	95.3	94.7	**95.4**
Wine	84.8	87.0	89.5	63.0	**94.4**
Glass	64.0	69.1	52.0	**81.5**	65.9
Flare	80.1	84.7	**85.0**	83.0	**85.0**
Pageblocks	92.3	**92.4**	89.0	91.0	91.0
Nursery	95,0	**95,4**	88,8	90,3	88,8

Table 8. Example

	B	$\bar{B}$	Total
A	201	49	250
$\bar{A}$	699	51	750
Total	900	100	1000

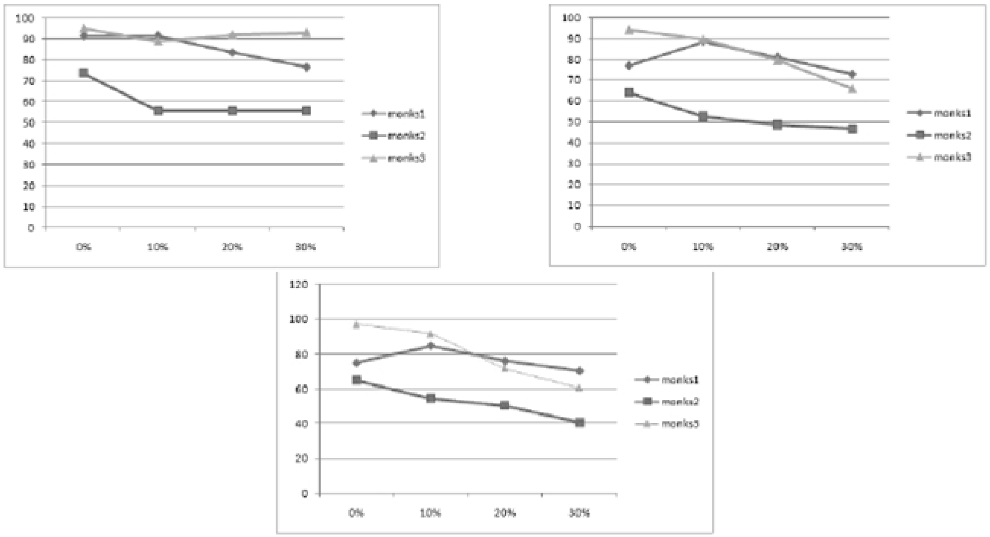


Fig. 4. Robustness Comparison Left: Our proposed approach Right: ID3 Approach Bottom: C4.5 approach

5. Conclusion

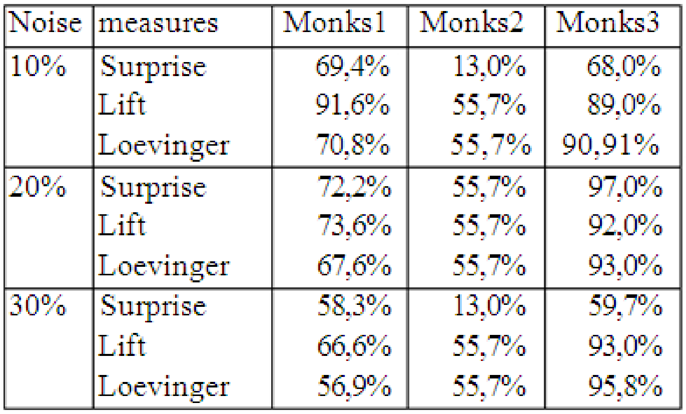
In this paper, we compared two versions of our proposed associative classification method that extracts directly a classifier formed by generic associative classification rules from a training set in order to reduce the number of associative classification rules without jeopardizing the classification accuracy.

Carried out experiments outlined that it extracts pertinent generic associative classification rules with small premises and gives competitive accuracies in comparison with popular classification methods. Further- more, experiments show that our classifier can be considered as robust in comparison with studied approaches.

In the near future, we will investigate new metrics for the rule selection.

We will extract multiple class labels using association rule discovery and apply our approach to a wide range of applications like text categorization and biological applications.

Table 9. Accuracy vs Correlation Measures



Noise	measures	Monks1	Monks2	Monks3
10%	Surprise	69,4%	13,0%	68,0%
	Lift	91,6%	55,7%	89,0%
	Loevinger	70,8%	55,7%	90,91%
20%	Surprise	72,2%	55,7%	97,0%
	Lift	73,6%	55,7%	92,0%
	Loevinger	67,6%	55,7%	93,0%
30%	Surprise	58,3%	13,0%	59,7%
	Lift	66,6%	55,7%	93,0%
	Loevinger	56,9%	55,7%	95,8%

References

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In J. B. Bocca, M. Jarke, and C. Zaniolo, editors, *Proceedings of the 20th Intl. Conference on Very Large Databases, Santiago, Chile*, pages 478-499, June 1994.

[2] M. Antonie and O. Zaiane. Classifying Text Documents by Associating Terms with Text Categories . In *Proceedings of the Thirteenth Austral-Asian Database Conference (ADC'02)*, Melbourne, Australia, 2002.

[3] M. Antonie and O. Zaiane. Text Document Categorization by Term Association. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'2002)*, pages 19-26, Maebashi City, Japan, 2002.

[4] Y. Bastide, N. Pasquier, R. Taouil, L. Lakhal, and G. Stumme. Mining minimal non-redundant association rules using frequent closed itemsets. In *Proceedings of the International Conference DOOD'2000, LNAI, volume 1861, Springer-Verlag, London, UK*, pages 972-986, July 2000.

[5] I. Bouzouita, S. Elloumi, and S. BenYahia. Garc : a new associative classification approach. In A. M. Tjoa and J. Trujillo, editors, *Pro- ceedings of 8th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2006), Springer-Verlag, LNCS 4081, Krakow, Poland*, pages 554-565, 4-8 September 2006.

[6] B. Ganter and R. Wille. *Formal Concept Analysis*. Springer-Verlag, 1999.

[7] J. Han and M. Kamber. *Data Mining : Concepts and Techniques.* Morgan Kaufmann., 2001.

[8] S. Lallich and O. Teytaud. Evaluation et validation de l'int´ ˆ des eret r` egles d'association. In *RNTI-E*, pages 193-217, 2004.

[9] W. Li, J. Han, and J. Pei. CMAR: Accurate and efficient classification based on multiple class association rules. In *Proceedings of IEEE International Conference on Data Mining (ICDM'01)*, pages 369-376, San Jose, CA, 2001. IEEE Computer Society.

[10] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *Knowledge Discovery and Data Mining*, pages 80-86, 1998.

[11] J. Quinlan and R. Cameron-Jones. FOIL: A midterm report. In *Proceedings of European Conference on Machine Learning, Vienna, Austria*, pages 3-20, 1993.

[12] J. R. Quinlan. C4.5 : Programs for Machine Learning. 1993.

[13] J. Wang and G. Karypis. HARMONY: Efficiently mining the best rules for classification. In *Proceedings of the International Conference of Data Mining*, pages 205-216, 2005.

[14] J. H. Xiaoxin Yin. CPAR: Classification based on Predictive Association Rules. In *Proceedings of the SDM*, pages 369-376, San Francisco, CA, 2003.

[15] O. Zaiane and M. Antonie. On pruning and tuning rules for associative classifiers. In *Proceedings of Ninth International Conference on Knowledge Based Intelligence Information And Engineering Systems (KES'05)*, pages 966-973, Melbourne, Australia, 2005.