© REVISTA DE MATEMÁTICA: TEORÍA Y APLICACIONES 2020 **27**(2) : 333–354
CIMPA – UCR ISSN: 1409-2433 (PRINT), 2215-3373 (ONLINE)
DOI: https://doi.org/10.15517/rmta.v27i2.42438

ASSOCIATIVE CLASSIFICATION WITH MULTIOBJECTIVE TABU SEARCH

CLASIFICACIÓN ASOCIATIVA CON BÚSQUEDA TABÚ MULTIOBJETIVO

RICARDO P. BEAUSOLEIL*

Received: 25/06/2019; Revised: 11/11/2019; Accepted: 06/02/2020

*Instituto de Cibernética Matemática y Física, Departamento de Matemática Interdisciplinaria, La Habana, Cuba. E-Mail: rbeausol@icimaf.cu

Abstract

This paper presents an application of Tabu Search algorithm to association rule mining. We focus our attention specifically on classification rule mining, often called associative classification, where the consequent part of each rule is a class label. Our approach is based on seek a rule set handled as an individual. A Tabu search algorithm is used to search for Pareto-optimal rule sets with respect to some evaluation criteria such as accuracy and complexity. We apply a called *Apriori* algorithm for an association rules mining and then a multiobjective tabu search to a selection rules. We report experimental results where the effect of our multiobjective selection rules is examined for some well-known benchmark data sets from the UCI machine learning repository.

Keywords: combinatorial data analysis; associative classification; tabu search; multiobjective optimization.

Resumen

Este artículo presenta una aplicación de Búsqueda Tabu Multiobjetivo a la minería de reglas de asociación. Centramos nuestra atención específicamente en la minería de reglas de clasificación, frecuentemente llamada clasificación asociativa, donde la parte consecuente es una clase. Nuestro enfoque se basa en la búsqueda de un conjunto de reglas manipulado como un individuo para la clasificación. Un algoritmo de Búsqueda Tabu es utilizado para encontrar conjuntos de reglas Pareto-Óptimo con respecto a algunos criterios tales como exactitud y complejidad. Aplicamos el siguiente algoritmo de *A priori* para la extracción de las reglas de asociación del problema en cuestión y entonces una búsqueda Tabu multiobjetivo es realizada para seleccionar subconjuntos de reglas. Reportamos experimentos donde es examinado el efecto de la selección multiobjetivo para algunos conjuntos de datos bien conocidos de la base de datos del almacén de máquinas de aprendizaje de la UCI.

Palabras clave: análisis de datos combinatorio; clasificación asociativa; búsqueda tabú; optimización multiobjectivo.

Mathematics Subject Classification: 90C27, 90C29, 90C30, 90B50, 93B40.

1 Introduction

Data mining is a very active and rapidly growing research area that involves pattern recognition technologies, as well as statistical and mathematical techniques. The task of data mining is to extract useful knowledge for human users from a database. Association rule mining is one of the most well-known data mining techniques [1], in this we seek to uncover associations among the attributes containing the database. The application of association rule mining to classification problems is often referred to as classification rule mining or associative classification [12], [13], [15], and [16], being applied in different areas like: finance, creditworthiness of clients, performance analysis of finance, investments, health care/medicine and others. Classification rule mining usually consists of 3 phases: Rule discovery, Rule selection and Classification. In the rule discovery phase, a large number of classification rules are extracted from a database using an association rule mining technique. All classification rules satisfying the minimum support and confidence are usually extracted from a database. A part of extracted classification rules are selected to design a classifier in the rule selection phase using a heuristic rule sorting criterion and the unlabelled data will be classified in the third step. The accuracy of the designed classifier usually depends on the specification of the minimum support and confidence and also, a threshold selected to subtract candidates to classification rules. Besides the support and confidence, other measures have been defined to measure the goodness of an association rule. Among them are gain, variance, chi-squared value, entropy gain, Laplace, lift, and conviction [2]. It is shown in [2] that the best rule according to any of the above mentioned measures is a Pareto-Optimal rule with respect to support and confidence. Evolutionary algorithms were proposed to search for Pareto-Optimal classification rules with respect to support and confidence for partial classification [7],[9]. Evolutionary algorithms were also used to search for Pareto-Optimal rule sets in classification rule mining [10], [11] where the accuracy of rule sets was maximized and their complexity was minimized. In this paper, we empirically examine the effect of multiobjective rule selection through computational experiments for some well known benchmark data sets from the UCI machine learning repository.This paper is organized as follows: First we explain some basic concepts in associative classification mining in Section 2. Section 3 explains the associative classification in a multiobjetive framework; in Section 4 we report experimental results on some well-known benchmark data sets. Conclusions in the Section 5.

2 Associative classification

Let D be a training data set with m attributes $A_1, A_2, \ldots, A_m$ and $|D| = n$ instances. C is a class label. Then, the values of attribute A_j and class C can be noted as a_j and c, respectively. An item set denoted by *itemset* is a set of several attribute values $< a_{i1}, a_{i2}, \ldots, a_{iq} >, q \leq m,\ 1 \leq i \leq n$ while an instance is $(a_1, a_2, \ldots, a_m, c)$. A classification rule $r =< a_{i1}, a_{i2}, \ldots, a_{iq}, c > q \leq m$

is a combination of an item set and a class label. For our pattern classification problem, we use classification rules of the following type:

$$r = \text{ if } \; a_{i1} \wedge a_{i2} \wedge \ldots \wedge a_{iq} \; \text{ then } \; c.$$

The statement $< a_{i1} \wedge a_{i2} \wedge \ldots \wedge a_{iq} >$ is the antecedent and the label *C* is called the consequent. *Occr(r)* is the number of instances in *D* that match the item set (antecedent condition) of *r*. *Suppcount(r)* is the numbers of instances in *D* that match the item set and the class label of *r* as well. Thus the support of *r* is $Supp(r) = Suppcount(r)/|D|$ while the confidence of *r* is $Conf(r) = Suppcount(r)/Occr(r)$. Them *minsupp* and *minconf* is the thresholds of support and confidence of rules given by user. Set *R* is the collection of classification rules where $\forall r \in R$ satisfies $Supp(r) > minsupp$ and $Conf(r) > minconf$. Given $r =< a_{i1}, a_{i2}, \ldots, a_{iq}, c >$ it implicates that if the attribute values of an instance match $a_{i1}, a_{i2}, \ldots, a_{iq}$ completely, the confidence of this instance belonging to class *C* is *Conf(r)* . Associative classification is to collect rules in training data set *D*, organize them in a certain order to form a classifier, denoted by *Cl*. When provided an unlabelled object, the classifier selects the rule in accordance with the order whose condition matches the objects and assigns class labels of the rule to it.

2.1 Reduced classifier

We simplify the classifier reducing the number of redundant rules. First, the rules are ordered in descending order, taking into account the following ranking rule: Given the rules r_1 and r_2:

If $Supp(r_1).Conf(r_1) > Supp(r_2).Conf(r_2)$, then $r_1 \prec r_2$, noted r_1 precedes r_2. An item set is denoted by $itemset_i$. Then r_2 is redundant if:

$$r_1 =< itemset_1, c_k > \;\text{ and }\; r_2 =< itemset_1, c_p >, \text{ but } \; r_1 \preceq r_2,$$

$$r_1 =< itemset_1, c_k > \;\text{ and }\; r_2 =< itemset_2, c_k >, itemset_1 \subseteq itemset_2, \\ \text{and } \; r_1 \preceq r_2.$$

2.2 Mining classification rules

Briefly we explain the generation of classification rules based on the Apriori algorithm. The frequency of an $< itemset, c >$ is the number of instances that contain the particular $< itemset, c >$. The mining of association rules from large databases is a two-steps process:

1. Find all $< itemset, c >$ with frequency $\geq \psi$, a frequent $< itemset, c >$ is denoted by *freqitemset*.

2. From all *freqitemset*, generate association rules satisfying the minimum support and confidence conditions.

Next, we turn to the task of generating association rules using the *freqitemset*. The Apriori algorithm takes advantage of the priori property to reduce the search space. The Apriori property states that if an $< item, c >$ is not frequent, then adding another value of attribute to this $< itemset, c >$ will not make this new $< itemset, c >$ more frequent. Then the algorithm is accomplished using the following two-step process, for each *freqitemset* denoted by *s*:

1. First, generate all subsets of *s*.

2. Then, let *ss* represent a nonempty subset of *s*. Consider the association rule $r : ss \Rightarrow (s - ss)$, where $(s - ss)$ indicates the set *s* without *ss*. Generate (and output) *r* if *r* fulfills the minimum support and confidence requirement. Do so for every subset *ss* of *s*.

2.3 Discretizing numeric attributes

Association rules are often used when the attributes are binary, it can be extended to numeric attributes discretizing they in nominal ones. To do this we use a method based on a model that uses a psycho-physical law called the Weber's law, stating that the noticeable difference in stimulus intensity must be proportional to the actual stimulus intensity itself [14]. Briefly we explain the model, assume that each range (a_{min}, a_{max}) of the attributes can be subdivided into a number of subintervals then, the method takes the following gridpoints:

$$a_0 = a_{min} + \epsilon_0, a_1 = a_{min} + \epsilon_1, \ldots, a_i = a_{min} + \epsilon_i.$$

The $\epsilon_0, \epsilon_1, \ldots$ represent the echelons of the so-called category under construction. Based on the Weber's law we set:

$$\epsilon_i - \epsilon_{i-1} = \xi \epsilon_{i-1}.$$

From this we obtain:

$$\epsilon_i = (1 + \xi)\epsilon_{i-1} = (1 + \xi)^2 \epsilon_{i-2} = \ldots = (1 + \xi)^i \epsilon_0.$$

The echelons constitute a sequence with geometric progression. The initial step is ϵ_0 and $(1 + \xi)$ is the progression factor. The integer-valued parameter *i* is chosen to designate the order of magnitude of the echelons. If we take $(1+\xi) = p$ we can write $\epsilon_i = p^i \epsilon_0$, then $a_{max} = a_{min} + p^h \epsilon_0$. From this we can write:

$$a_i = a_{min} + (a_{max} - a_{min})p^i/p^h, i = 1, 2, \ldots, h.$$

Then, we can vary the integer-valued i in order to obtain different a_i grid-points and the progression factor p to obtain a variety of the respective geometric scales [14].

3 Multiobjective in mining classification rules

3.1 Multiobjective metaheuristics

Most of the problems of optimization that think about in the practice involve several approaches to be optimized and that they can be contradictory to each other. It happens this way to the classification problems, in which is sought to optimize as much the quality as the complexity of the classifiers, and the improvement of an approach usually brings the worsening of another. The problems of optimization with more than a function objective, they are known as Multi-objective Optimization (MOP, of their initials in English) at the present time many areas present this type of problems, reason why the field of the MOP it is attracting a great interest and it constitutes a branch of wide investigation. Usually, in the problems of MOP they cannot be solutions that optimize in a simultaneous way all the objectives. It is for it that instead of looking for an only good solution, he/she tries to be a group of feasible solutions that you/they maintain the best commitment among all the objectives of the problem, denominated good solutions Pareto. A solution is good Pareto if it is not possible to improve one of the objectives without it worsens some other one.

The group of good solutions Pareto cannot be ordered in a global way, since any solution in the Pareto set is better or worse that other (a classification doesn't exist in…). The identification of the solution of better commitment or balance among the approaches that are optimized, require keeping in mind the user's preferences (decision maker) on the functions objectives. The complexity of a MOP depends on the size of the problem, where they influence such factors as the quantity of involved objectives or the size of the search space. The time of resolution of the same one should be reasonable so that you can apply to real problems found in the practice. In this sense, from final of the years 80, it has been carried out a wide study and development of multiobjective metaheuristics. The main objective of the resolution of a multiobjective problem is to obtain the group of good solutions Pareto. Nevertheless, when a metaheuristic is applied in it's resolution, the objective becomes to obtain a good approach of this group. In the resolution of MOP's, due to the global non existence of a good one, it should talk to new ideas that differ of the existent ones in optimization mono-objective. For they are introduced it's new concepts and definitions that distinguish to the MO like a branch of independent study.

3.2 Basic concepts

A general multiple objective optimization (MOO) problem consists of optimizing a set of $r \geq 2$ objective functions. It can be formulated as follows:

$$minimize\{f(x) : f(x) = (f_1(x), f_2(x), \ldots, f_r(x))\},$$

s.t.

$$x \in X,$$

where a solution $x = (x_1, x_2, \ldots, x_n) \in X$ is represented by a vector of n decision variables, X is a set of feasible solutions.

The image of a solution x in the objective space is a point:

$$z = (z_1, z_2, \ldots, z_r) = f(x).$$

Having several objective functions, the notion of optimum changes. The aim here is to find a good compromise rather than a unique solution as in a single-objective optimization problem. A MOP problem obtains a set of solutions known as the Pareto optimal, related to the following concepts.

Definition 1 (Pareto Dominance) *A solution $x^1 \in X$ dominates another solution $x^2 \in X$ if and only if $\forall i \in \{1, 2, \ldots, n\}$, $f_i(x^1) \leq f_i(x^2)$, and $\exists j \in \{1, 2, \ldots, n\} : f_j(x^1) < f_j(x^2)$.*

Definition 2 (Efficiency) *A solution x^* is efficient if and only if there is not another solution $x \in X$ such that x dominates x^*.*

The whole set of efficient solutions is the Pareto Optimal set, and is denoted by X_P. The image of a Pareto Optimal set in the objective space results in a set of non-dominated vectors denoted by PF and called *nondominated set* or Pareto Frontier.

The aim in multiobjective metaheuristic optimization is to obtain a Pareto Optimal set or a good approximation to it. This is a very difficult task and it depends on the practical complexity of the problem. As we said above, the introduction of preference information permits us to narrow the search over the regions of interest of the decision-maker.

3.3 Multiobjective in the mining of classification rules

Multiobjective techniques in the mining of classification rules can be roughly categorized into two approaches. In one approach, each rule is evaluated according to multiple rule evaluation criteria such as support and confidence. In our approach an adaptive search algorithm is used to search for Pareto-Optimal classification rules sets, taking into account the accuracy and complexity. These rules are meant to be interpreted in order: the first one, then if it doesn't apply the second, and so on. This set of rules that are intended to be interpreted in sequence is called a decision list.

In this section, we explain our Tabu search algorithm to seek these rules sets. In our study, we use the following two objectives:

- $f_1(Cl)$: The number of correctly classified patterns.
- $f_2(Cl)$: The number of selected rules.

The first objective associated to the accuracy is maximized while the second objective associated to the complexity is minimized. The maximum number of items in the antecedents is fixed to four, because bigger quantity of antecedents hinders the understanding of the rule and the analysis for the decision making.

3.4 Multiobjective tabu search

Tabu Search (TS) was proposed in its present form by Glover [6]. TS can be described as an intelligent search that uses adaptive memory and responsive exploration. It is an iterative technique that explores a set of problem solutions, denoted by X, by repeatedly making moves from one solution x to another solution x_0 located in the neighbourhood $N(x)$ of x.

3.4.1 Neighbours

Let us assume that we have already extracted n classification rules in the rule discovery phase of classification rule mining. These n classification rules are used as candidate rules in the classification rule phase. Let Cl be a set of the $p \leq n$ candidate rules (i.e. Cl is a classifier). We use a binary string of length n to represent Cl where 1 and 0 mean the inclusion in Cl and the exclusion from Cl of the corresponding candidate rule. A neighbour is generated changing the value of one selected binary variable, in the case of diversification move, based in previous experience [3], we change $\lceil \sqrt{n}\, \rceil$ variables. The number of neighbours is equal to $\lfloor \sqrt{n} \rfloor$.

3.4.2 Memory structures

The main components of tabu search approaches are memory structures, in order to have a trace of the evolution of the search. TS maintain a selective history of the states encountered during the search, and replaces $N(x)$ by a modified neighbourhood $N^*(x)$.

In our implementation we use a recency memory so called tabu list T that takes account the identity of the elements that changed values and become tabu. In order to maintain the tabu list, one circular list of t length is used and the index j of the variable x_j that becomes tabu is added to T.

A complementary memory structure associated with the tabu list we have, the array *freqcount* showing the frequency distribution of the selected variables, that is, the number of time that each variables takes part in the visited solutions. Our algorithm uses this frequency information to select the variables fewer selected.

3.4.3 Moves

The algorithm begins an intensification phase with a unitary movement while potentially non dominated solutions are found, if during a certain number of continuous iterations they are not potentially non dominated solutions, then the algorithms spends to carry out a long movement to leave that environment to other fewer visited and immediately in the new environment another intensification phase is restarted.

To Move one variable

- To select one variable taking into account of the frequency memory.
- To change the value of the selected variable.

To Move more than one variable

- To select a quantity equal to $\lceil \sqrt{n} \rceil$ taking into account the use of frequency memory.
- To change the value of the selected variables.
- To return to the movement of one variable.

3.4.4 Additive function value

In order to measure the quality of the solution we propose to use in our tabu search approach an Additive Function Value (*AFV*) with weighting coefficients $\lambda_k,\ AFV(x^{'}) = \sum_{k\in\{1,2\}},\ \lambda_k(z^*_k - f_k(x^{'}))$, where:

$$\begin{aligned}\lambda_k &= 2 - \exp(-s_k),\\ s_k &= |(f_k(x^{'}) - z^*_k)/z^*_k|.\end{aligned}$$

3.4.5 Search by goals

Let S the trial solutions set. An aspiration level is used to obtain an initial set of solutions as follows: Without lost generality, let us assume that every criterion is minimized.

Let $\triangle f(x^{'}) = (\triangle f_1(x^{'}), \triangle f_2(x^{'}))$ where $\triangle f_k(x^{'}) = z^*_k - f_k(x^{'}),\ k \in \{1,2\}$. A goal is satisfied, permitting x to be accepted and introduced in **S** if $[\exists \triangle f_k(x^{'}) \geq 0]$ or $[\forall k \in \{1,2\} \triangle f_k(x^{'}) = 0]$.The point Z^* is updated by $\max(z^*_k, f_k(x^{'})),\ k \in \{1,2\}$.

3.4.6 Attributes data structure

We use a memory that records information about the attributes of the solutions generated in each neighbourhood. This data structure contains the value of the additive function value, the selecting rules, and the index of variable that classify the solution as tabu or not tabu. This data structure provides information to choice the best move in the current neighbourhood. A procedure so called *SearchBestCandidate* is implemented to do this. In it an *aspiration criterion* to override an tabu move is used to choice this move when it result in a solution better than any visited so far.

3.4.7 Cutoff rule

The stop of the process is executed if the following statement is true.

$$(globit = maxit)\ or\ (\textit{non new dominated solution})$$

where *globit* is the count of iteration of the multi start search and *maxit* is the maximun multi start, then to stop put *cutoff* = *true*.

3.4.8 Tabu search approach

Notations:
R and P are sets of non-dominated solutions,
x^* is the best solution in R taking into account the product of the value of the objective functions,
optimal is the the value of the best performance achieved by the classifiers, taking into account the product of the value of the objective functions,
goal=true if a new non-dominated solution was introduced in S,
moveone=true if only one binary variable is changed,
itabu is a count of the number of iteration in the Taboo routine,
fan is the number of neighbours that will be generated,
ifan is a count of the current number of neighbours generated,
tenure contain the length of the tabu list,
M^* is a big bi-dimensional vector,
x_{best} is the best solution achieved in the neighbourhood,
bestsolution contain the value of the best solution achieved in the current neighbourhood,
globalopt is the value of the best solution achieved in the current tabu search.
$x^* \in R$: x^**is the best solution taking into account the following expression*:

$$\sum_{i=1,2} \lambda_i f_i taking \lambda_1 = 0.8, \lambda_2 = 0.2.$$

MultiStart Tabu
1. Create Rules (Apriori Algorithm).
2. $n \longleftarrow$ *number of rules*.
3. $tenure \longleftarrow \min(7, \lceil\sqrt{n}\rceil + 1)$.
4. $cutoff \longleftarrow false$.
5. $fan \longleftarrow \max(3, \lceil\sqrt{n}\rceil)$.
6. $globalopt \longleftarrow M(big)$.
7. $x \longleftarrow 0($*initial point*$)$.
8. $globit \longleftarrow 0$.
9. While $cutoff = false$.
10. $globit \longleftarrow globit + 1$.
11. $Z^* \longleftarrow M^*\ (big)$.
12. Taboo(x).
13. *Extract* $P \subseteq S \cup R$ by applying Pareto Dominance relation.
14. $R \longleftarrow P$.
15. $x \longleftarrow x^* \in R$.
16. end of while.

Taboo Routine

1. $goal\_unsatisfied \longleftarrow 0$.
2. $itabu \longleftarrow 0$.
3. $moveone \longleftarrow true$.
4. While $goal\_unsatisfied < (b = 5)$.
5. Candidate(x).
6. Update the tabu list.
7. Set $x \longleftarrow x_{best}$(best candidate).
8. if $goal = true$ then.
9. $goal\_unsatisfied \longleftarrow 0$.
10. else.
11. $moveone \longleftarrow false$.
12. $goal\_unsatisfied \longleftarrow goal\_unsatisfied + 1$.
13. endif.
14. endwhile.

Candidate Routine

1. $ifan \longleftarrow 0$.
2. $bestsolution \longleftarrow M$ (big).
3. $goal \longleftarrow false$.
4. While $ifan < fan$.
5. $x^{'} \longleftarrow Move(x)$.
6. Create $Cl(x^{'})$.
7. $AFV(x^{'})$.
8. Update $AttributeDataStructure$.
9. SearchByGoals($x^{'}$).
10. $ifan \longleftarrow ifan + 1$.
11. if *aspiration_level* then goal $\longleftarrow true$.
12. endwhile.
13. $SearchBestCandidate$.
14. $z_k^* \longleftarrow z_k^{next}, k = \overline{1, 2}$.

Search By Goals$(x^{'})$

1. if *aspiration_level* = *true*.
2. $S \longleftarrow x^{'}$.
3. $z_k^{next} \longleftarrow \max\{z_k^*, f_k(x^{'})\},\ k = \overline{1, 2}$.
4. endif.

Search Best Candidate

1. The candidate is tabu and not exist another candidate non-tabu.
2. *if* $globalopt \geq AFV(x^{'})$.
3. $bestsolution \longleftarrow AFV(x^{'})$.
4. $globalopt \longleftarrow bestsolution$.
5. $x_{best} \longleftarrow x^{'}$.
6. *else*.
7. *if* $bestsolution > AFV(x^{'})$.
8. $bestsolution \longleftarrow AFV(x^{'})$.
9. $x_{best} \longleftarrow x^{'}$.
10. endif.
11. The candidate is non-tabu.
12. if it is the first candidate non-tabu.
13. $bestsolution \longleftarrow AFV(x^{'})$.
14. $x_{best} \longleftarrow x^{'}$.
15. *else*.
16. *if* $bestsolution > AFV(x^{'})$.
17. $bestsolution \longleftarrow AFV(x^{'})$.
18. $x_{best} \longleftarrow x^{'}$.
19. endif.
20. *if* $globalopt > AFV(x^{'})$*then* $globalopt \longleftarrow bestsolution$.

3.4.9 Generating categories for numeric attributes and classifiers

The Apriori algorithm deals with nominal attributes and cannot handles ones measured on numeric attributes. To use it, first each attribute must be partitioned into an small number of intervals. There exist some methods to discretize numeric attributes, in our case we use the approach explained in 2.3. To make this, before the learning algorithm takes place us discretize giving different values to the parameters $p \in J = \{2, 3, 4\}$ and $i = \overline{2, 3, \ldots, h}$ with $h = 4$.

Associative classification

1. Read Data Training.
2. $optimal \longleftarrow M$(big).
3. $i \longleftarrow 1$ count of subintervals.
4. repeat.
5. $i \longleftarrow i + 1$.
6. $j \longleftarrow 0$.
7. repeat.
8. $j \longleftarrow j + 1$.

9. Discretize with (i, p_j).
10. Generate Rules.
11. MultiStartTabu.
12. $optimal > optimalcand$.
13. $optimal \longleftarrow optimalcand$.
14. $curfactor \longleftarrow p_j$.
15. $cursub \longleftarrow i$.
16. $x_{optimal} \longleftarrow x^*$.
17. until $j = |J|$.
18. until $i = h$.
19. Read Data Test.
20. Discretize with $cursub, curfactor$.
21. Generate the Best Rules Set with $x_{optimal}$.
22. Test Classifier with the Data Test.
23. Output Best Classifier.

4 Computational experiments

To evaluate the effectiveness of our method, several experiments were carried out and comparisons with different methods performed, four of these are rule methods and one based on Bayes' theorem. We extracted classification rules with four or less antecedent conditions using a minimum support and confidence. This restriction is because is difficult for human users to intuitively understand long classification rule with many antecedents condition.

4.1 Methods

The proposed algorithm is compared with the following methods. All methods are implemented in WEKA library [18].

1. Decision Table: It build a simple decision table majority classifier.
2. JRip: Based on RIPPER algorithm for fast and effective rule induction.
3. PART: Built a partial decision tree using the J48 algorithm.
4. NaiveBayes: Based on Naïve Baye's theorem.
5. OneR: Is a 1R classifier with one parameter: the minimum bucket size for discretization.
6. J48: Based on an implementation of the C4.5 decision tree learner.

4.2 Data sets

We perform experiments with a well known benchmark from UCI Machine Learning and Intelligent Systems Database [5].

Table 1: Data Sets description: I = Instances, A = Attributes, C = Classes,T=type of attributes (n:nominal,r:real), th = threshold for the frequency of the *freqitemset* taked in our approach.

Data Set	I	A	C	T	th%
Molecular Biology	106	57	2	n	16
Wine	178	13	3	r	10
Glass	214	9	6	r	5
Statlog Hearth	270	13	2	r	15
Ionosphere	351	34	2	r	15
Statlog Vehicle	846	18	4	r	10
Statlog Australian	690	14	2	n/r	10
Yeast	942	8	10	r	5
Wine_Quality	1599	11	6	r	10
Car	1728	6	4	n	5
Abalone	4177	8	29	n/r	3
Musk	6597	166	2	r	10

Table 3 shows the performance (based on the average number of rules) of our approach versus the algorithms that emit rules of decision. As you can observe the best results were obtained with the JRip algorithm and our approach MTS. Observe that for big data bases our approach obtained small groups of rules, not behaving of this way the remaining algorithms.

Nevertheless, the rules generated by MTS have a maximum of three antecedents, not being of this way with the remaining algorithms, including the JRip. In the appendix, Figure 1 and Figure 2 represent the rules generated for one of the experiments on the data set Abalone. In Figure 2 the rules generated by JRip are shown.

4.3 MTS with automatic threshold

As a result of the previous experiments we design one approach with automatic threshold that facilitates the use of the algorithm that we propose. With this end, it is introduced in the part that corresponds to Associative Classification the code that appears highlighted next. Initially $th = 10$.

Table 2: Classification accuracy(%): Multiobjective Tabu Search = MTS, DT = Decision Table. In it the words Statlog and Molecular were omitted.

Data Set	MTS	JRip	DT	NBayes	PART	OneR	J48
Biology	67.5	73.1	70.2	**81.4**	75.9	72.2	74.9
Wine	86.4	88.8	86.7	**97.3**	88.1	72.1	87.9
Glass	53.5	65.0	61.4	48.7	**67.7**	53.0	**67.7**
Hearth	72.5	80.7	79.2	**85.2**	83.4	70.9	79.9
Ionosphere	83.8	**89.1**	87.2	79.5	88.9	81.8	87.2
Vehicle	57.9	63.9	**66.8**	37.6	**66.8**	44.7	67.3
Australian	86.7	86.7	83.3	77.9	86.1	85.4	**86.9**
Yeast	42.2	**56.3**	54.3	56.2	54.4	36.2	55.4
Wine_Quality	41.9	**48.9**	45.2	37.9	43.4	46.8	40.8
Car	71.5	76.6	76.7	73.4	**80.5**	71.2	73.7
Abalone	23.4	18.2	**25.5**	23.3	20.6	23.4	20.2
Musk	**82.6**	72.1	80.1	73.5	75.1	76.5	70.8

Associative Classification

1. Read Data Training.
2. $optimal \longleftarrow M$(big).
3. $i \longleftarrow 1$ count of subintervals.
4. repeat.
5. $i \longleftarrow i + 1$.
6. $j \longleftarrow 0$.
7. repeat.
8. $j \longleftarrow j + 1$.
9. Discretize with (i, p_j).
10. **if** $th \leq 15$ **then** $th = th + 5$ **else** $th = 5$.
12. Generate Rules.
11. MultiStartTabu.
12. $optimal > optimalcand$.
13. $optimal \longleftarrow optimalcand$.
14. ***curth*** $\longleftarrow$ ***th***.
15. $curfactor \longleftarrow p_j$.

To the code 21 is added the following code ***th*** $\longleftarrow$ ***curth***. As you can observe in the Table 4, both approaches show similar results. The quantity of rules generated also stayed in the same range shown for MTS.

To compare the obtained results taking in consideration the objective that

Table 3: Comparison complexity: The average of rules of the classifiers.

Data Set	MTS	JRip	DT	PART	J48
Biology	17	4	17	6	13
Wine	14	4	18	4	6
Glass	8	7	14	12	16
Statlog Hearth	10	5	17	12	15
Ionosphera	8	5	17	6	18
Statlog Vehicle	17	12	59	27	50
Statlog Australian	11	5	17	25	24
Yeast	11	11	61	27	50
Wine_Q	15	14	60	75	118
Car	5	30	296	45	89
Abalone	9	15	52	763	873
Musk	7	17	449	33	95

Table 4: Comparison accuracy(%): MTS vs MTS with Automatic Threshold.

Data Set	MTS	MTS_AT
Biology	67.5	69.1
Wine	86.4	88.5
Glass	53.5	55.0
Statlog Hearth	72.5	73.5
Ionosphere	83.8	84.6
Statlog Vehicle	57.9	60.4
Statlog Australian	86.7	85.3
Yeast	42.2	43.5
Wine_Q	41.9	48.5
Car	71.5	72.5
Abalone	23.4	20.5
Musk	82.6	82.2

measures the accuracy, the following empiric measure of efficiency that measures the fraction of our solution that covers to the best solution found by the used algorithms was applied,

$$e = \frac{f_1(x_{optimal})}{Z_{opt}},$$

where Z_{opt} is the best solution found. Table 5 shows evidence of the good performance of our approach taking into account the accuracy, observe that a great quantity of results is bigger than 0.8, near to one maximum level of efficiency.

Table 5: Comparison accuracy (e).

Data Set	MTS	MTS_AT
Molecular Biology	0.83	0.85
Wine	0.88	0.91
Glass	0.79	0.81
Statlog Hearth	0.85	0.86
Ionosphere	0.94	0.95
Statlog Vehicle	0.86	0.90
Statlog Australian	0.99	0.98
Yeast	0.75	0.77
Wine_Quality	0.85	0.99
Car	0.89	0.90
Abalone	0.92	0.81
Musk	1.00	0.99

5 Conclusion

As you can see, we have carried out computational experiments with a variety of datasets with nominal and/or real attributes, with great quantity of instances and attributes, as well as, different categories of the class label. The discussion of the results that were presented during the different experiments, shows that our approach is competitive with the comparison algorithms, taking into account the accuracy. Also, our algorithm presents less complex rules that most of these algorithms and easily understood by human persons.

A very important factor is the definition of the threshold. A high threshold eliminates great quantity of rules, on the other side, a small threshold would leave a great quantity of them consuming a great quantity of memory. Taking this into account and that each database has a different threshold, in general, the user is forced to carry out a previous investigation to obtain the best threshold. We present a version with automatic threshold that facilitates the use of the algorithm improving most of the previous results.

Funding

This research was funded by the Institute of Mathematics Cybernetic and Physics (ICIMAF), Havana, Cuba.

References

[1] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A.I. Verkamo, *Fast discovery of association rules*, in: U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth & R. Uthurusamy (Eds.) Advances in Knowledge Discovery and Data Mining, AAAI Press, Menlo Park CA. **17**(1996), No.3, 307–328.

[2] R.J. Bayardo, R. Agrawal, *Mining the most interesting rules*, Proc. of 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (1999) 145–154. Doi: `10.1145/312129.312219`

[3] R.P. Beausoleil, R. Montejo, *A study with neighborhood searches to deal with multiobjective unconstrained permutation problems*, Journal of Industrial and Management Optimization, **5**,(2009), No. 2, 193–216. Doi: `10.3934/jimo.2009.5.193`

[4] R.P. Beausoleil, *MOSS-II: Tabu/Scatter Search for nonlinear multiobjective optimization*, in: Z. Michalewicz & P. Siarry (Eds.) *Advances in Metaheuristics for Hard Optimization*, Natural Computing Series, Springer, Berlin Heidelberg, 2007, pp. 39–67. Doi: `10.1007/978-3-540-72960-0_3`

[5] C. Blake, E. Keogh, C.J. Merz, *UCI Repository of machine learning repository*, University of California, Irvine (1998). Available at `http://archive.ics.uci.edu/ml/index.php`

[6] F. Glover, *Future paths for integer programming and links to artificial intelligence*, Computers and Ops. Res. 13(1986), no. 5, 533–549. Doi: `10.1016/0305-0548(86)90048-1`.

[7] B. de la Iglesia, M.S. Philpott, A.J. Bagnall, V.J. Rayward-Smith, *Data mining rules using multi-objective evolutionary algorithm*, Proc. of 2003, Congress on Evolutionary Computation, Canberrra, Australia, **3**, 2003, pp. 1552–1559. Doi:`10.1109/CEC.2003.1299857`

[8] B. de la Iglesia, A. Reynolds, V.J. Rayward-Smith, *Developments on a multi-objective metaheuristic (MOMH) algorithm for finding interesting*

sets of classification rules, in: C.A. Coello Coello, A. Hernández Aguirre, E. Zitzler (Eds) Evolutionary Multi-Criterion Optimization, Lecture Notes on Computer Science 3410, Springer, Berlin, 2005, pp. 826–840. Doi:10.1007/978-3-540-31880-4_57

[9] B. de la Iglesia, G. Richards, M.S. Philpott, V.J. Rayward-Smith, *The application and effectiveness of a multi-objective metaheuristic algorithm for partial classification*, European Journal of Operational Research **169**(2006), no. 3, 898–917. Doi: 10.1016/j.ejor.2004.08.025

[10] H. Ishibuchi, S. Namba, *Evolutionary multiobjective knowledge extraction for high-dimensional pattern classification problems*, in: X. Yao et al. (Eds.) Parallel Problem Solving from Nature, Lecture Notes in Computer Science, 3242. Springer, Berlin, 2004, pp. 1123–1132. Doi:10.1007/978-3-540-30217-9_113

[11] H. Ishibuchi, Y. Nojima, *Accuracy-complexity tradeoff analysis by multiobjective rule selection*, Proc. of ICDM 2005, Workshop on Computational Intelligence in Data Mining, 2005, pp. 39–48.

[12] W. Li, J. Han, J. Pei, *CMAR: Accurate and efficient classification based on multiple class-association rules*, Proc. of 1st IEEE International Conference on Data Mining, San José CA, 2001, pp. 369–376. Doi: 10.1109/ICDM.2001.989541.

[13] B. Liu, W. Hsu, Y. Ma, *Integrating classification and association rule mining*, Proc. 4th International Conference on Knowledge Discovery and Data Mining, 1998, pp. 80–86.

[14] F.A. Lootsma, *Scale sensitivity in the multiplicative AHP and SMART*, Journal of Multi-Criteria Decision Analysis **2**(1993), no. 2, 87–110. Doi: 10.1002/mcda.4020020205

[15] S. Mutter, M. Hall, E. Frank, *Using classification to evaluate the output of confidence based association rule mining*, in: G.I. Webb, X. Yu (Eds.) Advances in Artificial Intelligence, Lecture Notes in Artificial Intelligence 3339, Springer, Berlin, 2004, pp. 538–549. Doi: 10.1007/978-3-540-30549-1_47

[16] F. Thabtah, P. Cowling, S. Hammoud, *Improving rule sorting, predictive accuracy and training time in associative classification*, Expert Systems with Applications, **31**(2006), no. 2, 414–426. Doi: 10.1016/j.eswa.2005.09.039.

[17] F. Thabtah, P. Cowling, Y. Peng, *MCAR: Multi-class classification based on association rule*, IEEE International Conference on Computer Systems and Applications, Cairo, Egypt, (2005), pp. 127–133. Doi: `10.1109/AICCSA.2005.1387030`

[18] I.H. Witten, E. Frank, M.A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edn. Morgan Kaufmann, San Francisco CA, 2011. Doi: `10.1016/C2009-0-19715-5`

Appendix

```
RMTS rules
(Length >0.322) . (Diameter >0.077) . (0.037 < Whole-weight <=0.107) --> Class =  8
(0.126 < Height <=0.377) . (0.014 < Shell <=0.039) --> Class =  9
(Length >0.322) . (0.037 < Whole-weight <=0.107) --> Class =  9
(Sexo =  I) . (Length >0.102) . (0.014 < Shell <=0.039) --> Class =  8
(Whole-weight >0.943) . (Shucked-weight >0.056) --> Class = 11
(Diameter >0.253) . (0.014 < Height <=0.042) . (Vicera-weight >0.029) --> Class = 11
(Diameter >0.253) . (Vicera-weight <=0.029) --> Class =  7
(Sexo =  I) . (Shucked-weight <=0.056) --> Class =  6
(0.042 < Height <=0.126) --> Class =  6
(Whole-weight >0.943) . (Shell >0.039) --> Class = 10
(0.042 < Height <=0.126) . (Vicera-weight <=0.029) --> Class =  6
(Length >0.322) --> Class =  7
(Sexo =  I) . (Length >0.102) --> Class =  6
(Diameter >0.253) . (0.019 < Shucked-weight <=0.056) --> Class = 11
--> Class =  9
Number of Rules 14


Correctly Classified Instances      24.610965
Incorrectly Classified Instances     75.389035
```

Figure 1: Rules generated by MTS on the Abalone data set.

(Shucked-weight <= 0.2515) and (Whole-weight >= 0.7865) and (Length <= 0.515) => Class=23 (4.0/1.0)

(Shell <= 0.017) and (Shucked-weight <= 0.007) and (Height >= 0.03) => Class=3 (7.0/2.0)

(Whole-weight <= 0.041) and (Diameter >= 0.15) => Class=3 (5.0/2.0)

(Whole-weight <= 0.1185) and (Vicera-weight <= 0.014) and (Height <= 0.04) and (Length >= 0.165) => Class=4 (12.0/2.0)

(Length <= 0.28) and (Shell <= 0.0215) and (Diameter >= 0.18) => Class=4 (10.0/2.0)

(Shucked-weight <= 0.06) and (Whole-weight <= 0.0565) and (Vicera-weight >= 0.0065) and (Vicera-weight <= 0.0145) and (Sexo = I) => Class=4 (16.0/5.0)

(Diameter <= 0.23) and (Diameter <= 0.165) and (Length >= 0.225) => Class=4 (3.0/0.0)

(Shell <= 0.0305) and (Shell >= 0.0305) => Class=4 (2.0/0.0)

(Diameter <= 0.24) and (Vicera-weight <= 0.024) and (Shucked-weight >= 0.04) => Class=5 (28.0/10.0)

(Shell <= 0.077) and (Length <= 0.285) and (Diameter <= 0.19) and (Vicera-weight >= 0.014) => Class=5 (25.0/9.0)

(Shell <= 0.1125) and (Sexo = I) and (Height <= 0.095) and (Shucked-weight >= 0.049) and (Shucked-weight <= 0.0695) => Class=6 (50.0/23.0)

(Shell <= 0.1185) and (Height <= 0.095) and (Shucked-weight >= 0.0925) and (Sexo = I) and (Length >= 0.4) and (Length <= 0.42) => Class=6 (18.0/5.0)

(Shell <= 0.119) and (Sexo = I) and (Length <= 0.385) and (Shucked-weight >= 0.1) => Class=6 (49.0/24.0)

(Shell <= 0.119) and (Diameter <= 0.275) and (Shucked-weight >= 0.0735) and (Shell <= 0.0575) => Class=6 (38.0/16.0)

(Shell <= 0.172) and (Shell <= 0.133) and (Shucked-weight >= 0.147) and (Shell <= 0.114) and (Shell <= 0.094) => Class=7 (18.0/5.0)

(Diameter >= 0.485) and (Shucked-weight >= 0.615) and (Whole-weight <= 1.8325) and (Vicera-weight >= 0.375) => Class=11 (70.0/33.0)

(Shell <= 0.253) and (Sexo = I) and (Whole-weight >= 0.335) and (Whole-weight <= 0.625) and (Shucked-weight >= 0.2465) => Class=8 (78.0/35.0)

=> Class=9 (2806.0/2290.0)

Number of Rules : 18

Correctly Classified Instances	190	20.2559 %
Incorrectly Classified Instances	748	79.7441 %

Figure 2: Some rules generated by JRip on the Abalone data set.