*Proc. of International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA 2024)*
*1-2 February 2024, Victoria-Seychelles*

# Contribution of low support association rules in understanding the mined knowledge

Iztok Fister
*University of Maribor*
Maribor, Slovenia
iztok.fister@um.si

Andres Iglesias
*University of Cantabria*
Santander, Spain
iglesias@unican.es

Akemi Galvez
*University of Cantabria*
Santander, Spain
galveza@unican.es

Vili Podgorelec
*University of Maribor*
Maribor, Slovenia
vili.podgorelec@um.si

Iztok Fister Jr.
*University of Maribor*
Maribor, Slovenia
iztok.fister1@um.si

***Abstract*—In the realm of numerical association rule mining using population-based nature-inspired algorithms, the evaluation of results usually depends on a crucial metric – a fitness value crafted from weighted support and confidence measures. Traditionally, this metric separates the mined rules into a spectrum, ranging from the low-quality, low-fitness rules to the peaks of their high-quality, high-fitness counterparts. Until recently, little attention has been directed towards low-fitness rules, as users engage predominantly with high-quality rules. However, in our pioneering research, we go into the enigmatic realm of low-support, more precisely, the often-overlooked association rules. Through meticulous analysis of our rule repository, we seek to uncover the profound insights concealed within these seemingly unremarkable rules. Our method was applied to the Abalone UCI ML dataset, where three association rules, mined using the universal association rule miner based on evolutionary algorithms, were taken into consideration, i.e., the AR-1 was mined in the first generation, the AR-2 mined in the last generation, and the AR-3 mined at the moment before the phenomenon of the disappearing features has arisen. Significantly, the analysis of the AR-3 revealed that the association rules of the lower support can contribute to understanding the mined knowledge.**

***Index Terms*—association rule mining, data mining, evolutionary algorithms, numerical association rule mining, support**

## I. Introduction

Association Rule Mining (ARM) belongs to a class of Machine Learning (ML) algorithms [1]. The task of the ARM is to search for associations between attributes in a transaction database. Numerical Association Rule Mining (NARM) is an extension of canonical ARM which came into the foreground especially due to the advances in the research field of stochastic population-based search algorithms [2]. Most of the NARM rules are based either on Evolutionary Algorithms (EA)s or Swarm Intelligence (SI) algorithms. Here, the problem of association rule mining is packed into the optimization problem. uARMSolver [3] is an open source software for numerical association rule mining and implements the ARM-DE algorithm [4] for mining association rules [5].

Interestingly, the NARM algorithm mines a lot of association rules of lower fitness that are saved into an archive of mined association rules. The fitness value denotes the measure for estimating the quality of the association rule by the rule miner, and consists of a linear combination of NARM metrics. The archive presents an evolution of the mined association rules by the arbitrary algorithm for NARM from the history point of view, i.e., these association rules are evolved from the simple form of the lower fitness towards the more complex ones of the higher fitness.

The goal of the paper is to indicate what influence the low support (more precisely fitness function value) association rule for understanding the knowledge domain has by analyzing the archive of mined association rules. Thus, the simple association rules consist of fewer number of attributes, while the complex almost of all. The fitness value, that is a linear combination of three ARM measures (i.e., support, confidence and inclusion), converges to one by increasing the generation numbers. Unfortunately, some attributes are being lost from some association rules during the evolution process and replaced with others contributing more to the corresponding fitness values, while some numerical attributes being covered the whole domain of feasible values with the proposed interval. Indeed, those lost depends typically on the attribute type (i.e., categorical, real, and integer).

The following research questions are arose in the study:

- to identify the distribution of attributes within the transaction database,
- to detect the phenomenon of covering the whole domain of feasible values with intervals proposed by the NARM solver,
- to indicate the features being disappeared from the mining process.

Analyses of the exposed issues were performed on the Abalone dataset, because it is simple enough on the one hand (4177 transactions, nine features), and includes all feature types on the other. Let us note that the same analyses can be performed on any UCI ML dataset supporting mixed types of attributes.

In the remainder of the paper, the structure of the paper is as follows: Section II introduces the basic information necessary for readers to understand the subjects that follow.

This work was supported by the Slovenian Research Agency [Research Core Funding No. P2-0042 and No. P2-0057].

Corresponding author: Iztok Fister Jr. (e-mail: iztok.fister1@um.si).

979-8-3503-9452-8/24/$31.00 ©2024 IEEE

In Section III, the analyses of the Abalone dataset, dedicated for justifying the posted research questions, are described in detail. Discussions and outlining the future directions are part of Section IV. Section V summarizes the performed work.

## II. Basic information

### A. NARM

The NARM problem is defined mathematically as follows: Let us suppose a set of transactions $T = \{t_1, \dots, t_N\}$ is given, where each transaction is identified uniquely and contains a subset of features (also an itemset) $F = \{A_1, \dots, A_M\}$. The features can be either discrete or numerical (i.e., an integer or real). The discrete features can be drawn from a set of attributes $A^{(dis)} = \{a_1, \dots, a_Q\}$, while the numerical features can capture values from the interval $A^{(num)} \in [lb, ub]$, where the variables $lb$ and $ub$ designate the lower and upper bounds, respectively. Let us notice that the variable $M$ denotes the number of attributes, $N$ the number of the transactions in transaction database, and $Q$ the number of attributes. Then, an association rule is defined as an implication:

$$X \Rightarrow Y, \tag{1}$$

where $X$ and $Y$ are two itemsets and it holds that $X \cap Y = \emptyset$.

Several interestingness measures have been defined for identifying and evaluating the more important association rules in the literature. However, the most commonly used are support and confidence that are defined as follows:

$$supp(X \Rightarrow Y) = \frac{|t_i|t_i \in X \wedge t_i \in Y|}{N}, \tag{2}$$

$$conf(X \Rightarrow Y) = \frac{Support(X \cup Y)}{Support(X)}, \tag{3}$$

where $supp(X \Rightarrow Y) \geq S_{min}$ denotes the support and $conf(X \Rightarrow Y) \geq C_{min}$ the confidence of the association rule $X \Rightarrow Y$. Additionally, $S_{min}$ denotes minimum support and $C_{min}$ minimum confidence, determining that only those association rules with support and confidence higher than $S_{min}$ and $C_{min}$ are taken into consideration, respectively.

Additionally, an inclusion $incl(X \Rightarrow Y)$ NARM interestingness measure is defined as follows:

$$incl(X \Rightarrow Y) = \frac{ante(X \Rightarrow Y) + cons(X \Rightarrow Y)}{M}, \tag{4}$$

where $ante(X \Rightarrow Y)$ represents a set of objects belonging to the antecedent and $cons(X \Rightarrow Y)$ a set of objects belonging to the consequent. Mathematically, these functions are expressed as:

$$ante(X \Rightarrow Y) = \{o_{\pi_j} | \pi_j < Cp_i^{(t)} \wedge Th(Attr_{\pi_j}^{(t)}) = enabled\},$$
$$cons(X \Rightarrow Y) = \{o_{\pi_j} | \pi_j \geq Cp_i^{(t)} \wedge Th(Attr_{\pi_j}^{(t)}) = enabled\}.$$

Indeed, this measure estimates how many features contribute in the particular association rule among all. It is expressed as a real number in the interval $[0, 1]$. The closer this value is to one, the higher the inclusion, and vice versa.

### B. uARMSolver

Universal Association Rule Mining Solver (uARMSolver) [3] presents an open source framework for NARM written in C++. The framework consists of four parts: a problem definition, preprocessing, rule mining, and visualization. The problem is defined in the format supported by the UCI ML repository (i.e., .csv raw text) [6]. In the preprocessing stage, the uARMSolver transforms the raw text dataset into a transaction database, where more squashing methods are also available, in order to reduce the number of records by preserving the quality of the information. More EAs and SI-based algorithms can be used for NARM. At the moment, the Differential Evolution (DE) [7] and the Particle Swarm Optimization (PSO) [8] can be applied by the framework. The visualization part of the framework is open for including various modules for visualization.

The mined association rules are distinguished by different fitness function values. The fitness function has a crucial impact on the evolutionary search process. In uARMSolver, this is defined as a linear combination of NARM metrics, in other words:

$$f(\mathbf{x}_i^{(t)}) = \frac{\alpha \cdot supp(X \Rightarrow Y) + \beta \cdot conf(X \Rightarrow Y) + \gamma \cdot incl(X \Rightarrow Y)}{\alpha + \beta + \gamma}, \tag{5}$$

where $\alpha$, $\beta$, and $\gamma$ denote weights, $supp(X \Rightarrow Y)$, $conf(X \Rightarrow Y)$ and $incl(X \Rightarrow Y)$ represent the support, confidence and inclusion of the observed association rule, respectively.

As opposed to the Apriori, that stores each mined association rule to the archive regardless of its quality, the uARMSolver archives only those that outperform the current best ones. Consequently, the solver produces a lower number of association rules, on the one hand, but the archived ones are of better quality, on the other.

## III. Analysing an archive of mined association rules

An archive of mined association rules is analyzed in this section. The aim of this analysis was three-fold:

- To identify the distribution of attributes within the UCI ML dataset in order to determine the complexity of the problem.
- To detect a phenomenon of covering the whole interval of possible values by NARM solver.
- To indicate the problem of the features being disappeared in the association rules.

The archive of the mined association rules is built using a uARMSolver [3] that is available under the MIT licence. All experiments were performed on a HP OMEN desktop computer, with the following configuration:

- AMD Ryzen 7 5700G,
- HyperX 16GB Memory,
- NVIDIA GeForce RTX 3060 Ti,
- Linux Mint 21.2 Victoria.

All the analyses were performed on the Abalone dataset taken from the UCI ML repository [9]. The dataset is devoted for predicting the age of an abalone from its physical measurements. The age is determined by counting the number of rings through a microscope by a time-consuming procedure, during which the shell is cut using the cone and then it is strained. An easier method for determining the age is to use the other physical measures for this prediction as illustrated in Table I.

The dataset is simple enough for the complex analysis, because it consists of only 4,177 transactions, where each transaction consists of 9 features. The features are of three different types: categorical, continuous (also real), and discrete (also integer). Let us mention that the feature 'Rings' in Table I is the target, and, therefore, it presents the result of the classification.

### *A. Dataset Abalone as a set of random variables*

The distribution of attribute values in a transaction database has a crucial impact on the calculation of the NARM metrics. Therefore, it seems convenient to introduce a concept of random variables that serves as a basis for calculating how the probabilities are distributed over the random variable. The following deliberation is needed when the concept can be applied to the NARM: The transaction database is realized as a matrix $T$ of dimension $N \times M$, where the variable $N$ denotes the number of transactions and $M$ the number of features. Then, the database $T$ can be represented as a set of random variables $T = \{Z_1, \ldots, Z_M\}$, where the random variables $Z_i = \{z_1, \ldots, z_Q\}$ are referred to as features, and elements $\{z_i\}$ for $i = 1, \ldots, Q$ represents the frequency of occurrences of a particular feature belonging to specific sample points. Thus, the variable $Q$ limits the number of sample points.

A random (or stochastic) variable is defined as a random function that assigns a specific value to each point in the sample space [10]. If the variable is defined on a finite sample space, it is referred to as discrete, while the one defined on an infinite sample space is called a non-discrete random variable. The definition of the sample points depends on the feature type, as follows: When the type of random variable is discrete, the sample points are simply represented as the order number of discrete attributes, in other words:

$$\begin{aligned} S^{(dis)} &= \{1, \ldots, Q\}, \\ Z^{(dis)} &= \{z_1, \ldots, z_Q\}, \end{aligned} \tag{6}$$

where variable $S^{(dis)}$ denotes the order numbers of members of a discrete set $A^{(Dis)}$, and the elements $z_k \in Z^{(dis)}$ for $k = 1, \ldots, Q$ designate the frequency of occurrences of the discrete attributes from the transaction database $T$. On the other hand, when the feature type is numerical (i.e., integer or real-valued), the random variable is defined similarly as:

$$\begin{aligned} S^{(num)} &= \{1, \ldots, Q\}, \\ Z^{(num)} &= \{z_1, \ldots, z_Q\}, \end{aligned} \tag{7}$$

but the sample points are mapped to the equidistant intervals $k \mapsto [Lb_k, Ub_k]$ for $k = 1, \ldots, Q$ into which a whole interval of domain values $[LB, UB]$ is divided for a particular feature. Correspondingly, the random variable $z_k$ denotes the frequency of occurrences of the particular attributes within the $k$-th interval. Indeed, the variables $Lb_k$ and $Ub_k$ represent the lower and the upper bounds of the $k$-th equidistant interval, while the variables $LB$ and $UB$ are the lower and the upper bounds for the specific attribute in the transaction database. Thus, the $k$-th interval is determined as follows:

$$k = \left\lfloor \frac{Q \cdot (x - LB)}{UB - LB} \right\rfloor, \tag{8}$$

where the variable $x$ denotes the value of the corresponding attribute from the transaction database.

Table II illustrates the Abalone transaction database as a set of random variables with $Q = 10$ sample points, where each random variable reflects random characteristics of attributes belonging to a particular feature.

As evident from Table II, the frequencies of random variables are not normally distributed. Interestingly, the distributions have a different impact on the calculation of the NARM metrics: Each discrete feature consists of attributes $A^{(dis)}$, whose volume is limited by the size of the corresponding attribute set. On the other hand, the numeric features are limited with the interval $[lb, ub]$, as proposed by the NARM solver. Typically, the proposed interval captures more classes (i.e., equidistant intervals), while the NARM support and confidence metrics depend on the distribution of the included classes. For instance, the expected NARM support for the feature 'Length' is higher when the proposed interval comprises classes 7 and 8 than when the classes 1 and 2 are taken into consideration.

In common, the number of sample points by a random variable is defined as:

$$Q = \max(Q^{(num)}, \max_{\forall A^{(dis)} \in A}(|A^{(dis)}|), \tag{9}$$

where the number of sample points is the maximum value between the selected sample points by numerical features and those discrete features containing the highest number of attributes.

In general, the evolutionary search process mines the better association rules according to the fitness function values more easily, when there are more numeric features within the dataset, due to the flexibility of the intervals proposed by the NARM solver. On the other hand, when there are more discrete values arisen within the dataset, the searching for the better association rules is more complex by using the evolutionary search process.

### *B. How to detect covering the domain of feasible values by the numerical features?*

The characteristics of the uARMSolver are that it is capable of adapting the proposed interval of values by the numerical features towards the whole domain of feasible values. This means that we can expect that the proposed intervals will match the whole domain of feasible values by maturing the evolutionary search process. In this case, the values of support and confidence converge towards the value one.

TABLE I.
CHARACTERISTICS OF ABALONE DATASET FEATURES.

| Feature | Type | Unit | Domain | Note |
|---|---|---|---|---|
| 'Sex' | Categorical | n/a | {M,F,I} | Male, Female, and Infant |
| 'Length' | Continuous | mm | [0.0750,0.8150] | Longest shell |
| 'Diameter' | Continuous | mm | [0.0550,0.6500] | Perpendicular to length |
| 'Height' | Continuous | mm | [0.0000,1.1300] | With meat in the shell |
| 'Whole weight' | Continuous | g | [0.0020,2.8255] | Without abalone |
| 'Shucked weight' | Continuous | g | [0.0010,1.4880] | Weight of to meat |
| 'Viscera weight' | Continuous | g | [0.0005,0.7600] | After blending |
| 'Shell weight' | Continuous | g | [0.0015,1.0050] | After drying |
| 'Rings' | Integer | n/a | [1,29] | +1.5 age in years |

TABLE II.
ABALONE PRESENTED AS A SET IF RANDOM VARIABLES WITH $Q = 10$ SAMPLE POINTS.

| Feature | Class | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 'Sex' | 1,528 | 1,307 | 1,342 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 'Length' | 7 | 60 | 147 | 304 | 489 | 749 | 1,051 | 1,017 | 324 | 29 |
| 'Diameter' | 13 | 66 | 180 | 344 | 513 | 812 | 1017 | 934 | 275 | 23 |
| 'Height' | 1,023 | 3,129 | 23 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 'Whole weight' | 633 | 783 | 827 | 823 | 616 | 286 | 129 | 58 | 16 | 6 |
| 'Shucked weight' | 786 | 1,052 | 962 | 775 | 399 | 123 | 46 | 24 | 7 | 3 |
| 'Viscera weight' | 835 | 999 | 1,027 | 747 | 363 | 147 | 50 | 7 | 1 | 1 |
| 'Shell weight' | 777 | 1,023 | 1,078 | 798 | 349 | 104 | 33 | 9 | 5 | 1 |
| 'Rings' | 17 | 431 | 1,648 | 1,388 | 329 | 228 | 100 | 29 | 4 | 3 |

The purpose of the analysis was to detect the phenomenon of covering the whole domain of feasible values by numerical features. In line with this, two association rules were taken into consideration, i.e., the association rule AR-1 mined in the first (Table III), and the rule AR-2 mined in the last generation of the evolutionary search process (Table IV).

TABLE III.
STRUCTURE OF THE ASSOCIATION RULE AR-1 (32.79 %).

| Position | Feature | Attribute | Coverage | Total |
|---|---|---|---|---|
| Antecedent | 'Rings' | [8,16] | 28.57 % | 31.03 % |
| Consequent | 'Height' | [0.0000,0.3874] | 33.99 % | 33.66 % |
| | 'Sex' | 'F' | 33.33 % | |

Tables III-IV illustrate all the features with their corresponding attributes (i.e., intervals by numerical features) joined into the antecedent and consequent parts of the association rule. Thus, the column 'Coverage' reflects the percentage of domain values covered by the definite attribute, while the column 'Total' depicts the average coverage of attributes in either the antecedent or consequent, respectively. The percentage written in the title of the corresponding table presents the weighted average of the attributes in the association rule.

As can be seen from Table III, the AR-1 consists of one attribute in the antecedent and two attributes in the consequent. The total coverage of the attributes in the AR-1 is 32.79 %. Interestingly, both types of attributes, i.e., numerical and discrete, are included in the AR-1.

As evident from Table IV, the AR-2 consists of seven attributes in the antecedent and one attribute in the consequent. Interestingly, all the attributes are of the numerical type, while

TABLE IV.
STRUCTURE OF THE ASSOCIATION RULE AR-2 (42.86 %).

| Position | Feature | Attribute | Coverage | Total |
|---|---|---|---|---|
| Antecedent | 'Diameter' | [0.4572,0.6500] | 32.37 % | 46.65 % |
| | 'Height' | **[0.0000,1.1300]** | 100.00 % | |
| | 'Length' | [**0.0750**,0.3227] | 32.04 % | |
| | 'Rings' | [**1**,3] | 10.44 % | |
| | 'Shell weight' | [0.1657,0.4480] | 28.14 % | |
| | 'Viscera weight' | **[0.0005,0.7600]** | 100.00 % | |
| | 'Whole weight' | [0.5445,1.1844] | 22.67 % | |
| Consequent | 'Shucked weight' | [0.7094,0.9514] | 16.3 % | 16.33 % |

their attributes cover almost 43 % of the domain of feasible values. We can observe that even two proposed intervals (i.e., features 'Height' and 'Viscera weight') match the whole domain, and, thus, they achieved a covering of 100 %. Obviously, the covering increases the value of the support metric to one.

### *C. How to detect the feature of being disappeared?*

The phenomenon arises, when a specific feature disappears from the association rules being mined by the uARMSolver, due to a too small value of support or confidence. Therefore, the goal of the analysis was to find where the phenomenon is arising and how these disappearing features can contribute in understanding the knowledge hidden in the transaction database.

Typically, the disappearing features are of discrete type, because their supports metrics are limited with the number of different classes, into which these attributes can be classified. The numerical features have no such limitation, because their support metrics could converge to the value of one by widening of the proposed intervals by the uARMSolver.

In line with this speculation, we need to identify the most interesting association rule, where the feature being disappeared. Consequently, the archive of the association rules is sorted according to the fitness function values, i.e., the best association rules are at the beginning and the worst at the end of the archive. In that case, the problem of detecting the feature being disappeared can be defined inversely as: How to detect such association rule, where the discrete attribute arises at first.

The example of the association rule AR-3, including the feature being disappeared, is illustrated in Table V. The table is formatted similarly as Tables III-IV. As indicated in Table V, the AR-3 consists of six attributes in the antecedent and two attributes in the consequent. This means that the evolutionary search process cannot improve the support metric of the discrete feature 'Sex', and, consequently, this will be replaced by a more appropriate numerical feature in the next generations. Obviously, the AR-3 can substitute the knowledge about Abalone domain that is referred to as the discrete attribute 'Sex_I'.

TABLE V.
STRUCTURE OF THE ASSOCIATION RULE AR-3 (28.59 %).

| Position | Feature | Interval | Coverage | Total |
|---|---|---|---|---|
| Antecedent | 'Diameter' | [0.1366,0.2500] | 19.04 % | 29.60 % |
| | 'Length' | **[0.1136,0.4846]** | 49.60 % | |
| | 'Sex' | 'I' | 33.33 % | |
| | 'Shell weight' | [0.1226m,0.2033] | 8.05 % | |
| | 'Viscera weight' | [0.3224,**0.7600**] | 57.62 % | |
| | 'Whole weight' | [1.3499,1.6307] | 9.95 % | |
| Consequent | 'Rings' | [12,21] | 34.48 % | 25.56 % |
| | 'Shucked weight' | [1.2407,**1.4480**] | 16.63 % | |

## IV. DISCUSSION AND FUTURE DIRECTIONS IN THE EVOLUTION OF THE NARM ALGORITHM

The characteristics of the EAs are to improve the population of solutions according to the value of the fitness function from generation to generation. In uARMSolver, the fitness function is composed of three terms representing different NARM measures, i.e., support, confidence, and inclusion.

The NARM support metric is defined as a probability of the attribute occurring in the transaction within the transaction database. This means that this probability is proportional to the inverse value of the corresponding number of classes into which the discrete feature is divided (i.e., $\propto \frac{1}{|A^{(dis)}|}$). However, when the probability of occurring is equal by all discrete attributes, then the support metric usually cannot exceed the value of $1/|A^{(dic)}|$.

In the case of the numeric features, the support metric depends on the width of the interval, with which the different classes of equidistant intervals are captured. The more classes captured by the attribute, the higher is the support metric. The confidence metric depends indirectly on the support in the following sense: While the support measures the occurrence of the feature in the association rule regardless of whether it occurs in the antecedent or consequent, the confidence takes care where the feature arises.

Obviously, the best fitness value is obtained by the association rules with the maximum number of features, and the higher values of support and confidence measures. In general, the association rules with a smaller number of features, and lower values of support and confidence are mined at the beginning of the evolutionary search process (AR-1 in Table VI). Then, the association rules with a higher number of features and higher values of support and confidence are more appropriate (AR-3 in Table VI). Finally, the association rules with the full number of features and values of support and confidence closer to one are expected (AR-3 in Table VI).

TABLE VI.
SUMMARY OF OBSERVED ASSOCIATION RULES.

| ARs | Support | Confidence I | nclusion C | overage | Fitness |
|---|---|---|---|---|---|
| AR-1 | 0.3129 | 0.3129 | 0.3333 | 32.79 % | 0.3197 |
| AR-2 | 0.9998 | 1.0000 | 0.8889 | 42.86 % | 0.9629 |
| AR-3 | 0.3213 | 1.0000 | 0.8889 | 28.59 % | 0.7367 |

In order to automate the manual process of searching for the characteristic association rules AR-1, AR-2, and AR-3. the pseudo-code of the Interesting Association Rule Detector (IARD) algorithm was devised as illustrated in Algorithm 1. As Algorithm 1 indicates, the algorithm starts with the archive of the association rules sorted descendingly according to their fitness values. The task of the algorithm is to find three interesting association rules: (1) the worst AR-1, (2) the best, and (3) the rule AR-3 with the feature of being disappeared. These association rules are obtained by a straightforward walk through the archive of the sorted association rules.

**Algorithm 1** IARD algorithm for detecting the interesting ARs.

**Require:** $Ar$ - archive of the ARs sorted descendingly
**Ensure:** AR-1, AR-2, AR-3 - interesting ARs
  AR-1 $= \emptyset$, AR-2 $= \emptyset$, AR-3 $= \emptyset$
  AR-1_$full$ $=$ AR-2_$full$ $=$ **false**
  **for all** $r \in Ar$ **do**
    **if not**AR-2_$full$ **then**
      AR-2 $= r$ ▷ The best AR
      AR-2_$full$ $=$ **true**
    **end if**
    **for all** $a \in r$ **do**
      **if not**AR-3_$full$ **and** $a.type = categorical$ **then**
        AR-3 $= r$ ▷ The best discrete AR
        AR-3_$full$ $=$ **true**
        **continue**
      **end if**
    **end for**
  **end for**
  AR-1 $= r$ ▷ The worst AR

As evident from the performed analyses, the existing NARM algorithm suffers from a lack of competent handling with numerical attributes. Indeed, the problem is how to limit the convergence of the support metric towards one by the numerical attributes, and, thereby, not to degrade the evolutionary search process. The trivial solution of the problem is to introduce the limitations for excessive expansion of the intervals proposed by the NARM solvers. Obviously, there an additional issue emerged, i.e., how to set the limitation properly. The underestimated ranges could prefer emerging too many discrete attributes in the association rules, while the overestimated ranges could cause the discrete attributes to vanish from the association rules. In our opinion, the best solution is to consider the probability distribution of the attributes in the transaction database during the evolution process such that both types of attributes would have equal chances to be included into the mined association rules.

## V. Conclusion

Typically, the nature-inspired NARM solvers (e.g., uARM-Solver) produce a large archive of association rules that are distinguished according to their fitness function values from the lower of the worse rule quality to the higher of the better rule quality. The purpose of the study was to analyze if some additional knowledge can be explored from the association rules of lower quality.

The analyses performed on the Abalone UCI ML dataset showed that the NARM metrics (i.e., support and confidence) depend on the distribution of attributes within the transaction database. Furthermore, the phenomenon of covering the whole domain of feasible values with the intervals proposed by the NARM solver was detected that causes some features of being disappeared from the mining process. Finally, we can conclude that the association rules of lower support can contribute in understanding the mined knowledge, especially, those that include the features being disappeared from the mining process. Consequently, the algorithm for detecting the interesting association rules of lower quality was developed, based on the findings of the study.

The results of the analyses affect the further development of the NARM solvers crucially. At first, the intervals of numeric attributes as proposed by the NARM solvers should be limited by considering the probability distribution of attributes within the transaction database. In line with this, the analysis of the probability distributions could be primarily presented as a potential direction for future research.

## Acknowledgment

This work was supported by the Slovenian Research Agency [Research Core Funding No. P2-0042 and No. P2-0057]. A. Galvez and A. Iglesias thank the financial support from the projects PDE-GIR (Marie Sklodowska-Curie grant agreement No 778035) of the European Union's Horizon 2020 research & innovation program, and #PID2021-127073OB-I00 from the Agencia Estatal de Investigacion, Spanish Ministry of Science and Innovation (Computer Science National Program) and European Funds EFRD (AEI/FEDER, UE).

## References

[1] T. M. Mitchell, *Machine learning*. McGraw-hill New York, 1997, vol. 1, no. 9.

[2] E. Varol Altay and B. Alatas, "Performance analysis of multi-objective artificial intelligence optimization algorithms in numerical association rule mining," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 3449–3469, 2020.

[3] I. Fister and I. Fister Jr, "uarmsolver: A framework for association rule mining," *arXiv preprint arXiv:2010.10884*, 2020.

[4] I. Fister, A. Iglesias, A. Galvez, J. Del Ser, E. Osaba, and I. Fister, "Differential evolution for association rule mining using categorical and numerical attributes," in *Intelligent Data Engineering and Automated Learning–IDEAL 2018: 19th International Conference, Madrid, Spain, November 21–23, 2018, Proceedings, Part I 19*. Springer, 2018, pp. 79–88.

[5] v. Stupan and I. Fister Jr., "Niaarm: A minimalistic framework for numerical association rule mining," *Journal of Open Source Software*, vol. 77, no. 7, pp. 44–48, 2022.

[6] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[7] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[8] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948 vol.4.

[9] W. e. a. Nash, "Abalone," UCI Machine Learning Repository, 1995, DOI: https://doi.org/10.24432/C55C7W.

[10] J. Schiller, R. Srinivasan, and M. Spiegel, *Schaum's Outline of Probability and Statistics*, ser. McGraw-Hill's ebook library: Student study aids. McGraw-Hill Education, 2013.