Toward Reusing the Numerical Association Rule Mining Models

Iztok Fister Jr.[1(✉)], Andres Iglesias[2,3], Akemi Galvez[2,3], and Iztok Fister[1]

[1] Faculty of Electrical Engineering and Computer Science, University of Maribor, Smetanova 17, 2000 Maribor, Slovenia
iztok.fister1@um.si

[2] University of Cantabria, Avenida de los Castros, s/n, 39005 Santander, Spain

[3] Toho University, 2-2-1 Miyama, Funabashi 274-8510, Japan

Abstract. Nowadays, it is no secret that modern machine learning methods are amongst the more computationally-intensive learning methods. The rise in the applications of computationally-intensive deep learning, automated machine learning methods, and even metaheuristics for optimization, have increased the consumption of electrical energy dramatically. Consequently, we can predict that the numbers of global carbon footprints, arising as a byproduct of the increased consumption of electrical energy during intensive computation, will be higher and higher in the near future. Fortunately, the research community is aware of these problems, and is, thus, looking for solutions of how to reduce the carbon footprint in the sense of the so-called Green AI. In line with this, the paper introduces a reusable model for Numerical Association Rule Mining which is also one of the hardest computationally-intensive learning methods. In the classical Numeric Association Rule Mining, the model (i.e., the archive of mined association rules) is created anew, when the new incoming association rule are emerging. The proposed reusable model only modifies the existing model accordingly by incoming new rules, while the complexity of computation decreases remarkably.

Keywords: Association rule mining · Numerical association rule mining · Modeling · Metaheuristics · Green AI

1 Introduction

Nowadays, Artificial Intelligence (AI) is presented almost everywhere for helping humans in making decisions. We can easily mention that AI methods can be found in various environments from logistics, through Industry 4.0, and even to medicine. The AI models have already advanced to such an extent that they are often able to beat the human in solving particular tasks. However, these advancements are not free of charge. In this case, the accuracy of a machine learning model is associated with the consumption of electrical energy needed for computing. More accurate, efficient and reliable models require a lot of electrical

H. Sanjurjo González et al. (Eds.): SOCO 2021, AISC 1401, pp. 198–206, 2022.
https://doi.org/10.1007/978-3-030-87869-6_19

energy in the training phases, such as Neural Architecture Search or Hyperparameter optimization. The energy consumption of these methods can also be measured by a measure of the carbon impact of AI [3]. Recently, a very interesting perspective on this domain was published in [8], where the authors coined the term **Red AI** that exposes those AI methods which are environmentally unfriendly [10]. When AI methods are environmentally friendly, they are referred to as the **Green AI** [8].

Fortunately, the research movement is also becoming aware of the carbon impact of AI, and has put a lot of effort into the development of training AI models such that they do not need a lot of re-training in order to save computational power as well as computational electrical energy. Potentially, they also save much money when reducing the computational power of the dedicated hardware or clouds they use.

In this paper, we take a look on Numerical Association Rule Mining (NARM) which is a very popular method in Machine Learning (ML). The task of the NARM is to discovery regularities between objects within large-scale transaction databases. These regularities are presented in the sense of the association rules. In NARM, all objects consist of numerical attributes that are determined by intervals of feasible values. Especially, it is well-suited for solving with the nature-inspired metaheuristics, which encompass two algorithm families [7]: Evolutionary Algorithms (EAs) [5] and Swarm Intelligence (SI) algorithms [2]. This study focuses on Differential Evolution [9], which is also suitable for NARM solving [6].

Typically, the NARM problem is solved offline by generating an archive of mined association rules. Thus, the archive that can be observed as the NARM model is generated anew. This method becomes critical when we start to work with big data. Here, we try to propose an online solution of the problem, when the new incoming transactions are saved into a transaction database and simultaneously affect the NARM model. As a result, the NARM model is not created anew, but reused. The proposed algorithm, complying with the Green AI, was applied to the UCI ML dataset repository, while the obtained results are promising, and encourage us to continue with the work in the same direction. As far as the authors' knowledge, this point of view was not treated for NARM by the research community until now.

The structure of the paper is as follows. Section 2 discusses the basics of the NARM. The proposed method for reusing the NARM model is the subject of Sect. 3. The practical simulations and evaluations of obtained results are illustrated in Sect. 4. The paper finishes with Sect. 5, where the results are summarized and the directions are outlined for the future work.

2 Numerical Association Rule Mining

The ARM problem is defined formally as follows: Let us suppose a set of objects $O = \{o_1, \ldots, o_m\}$ and transaction database D are given, where each transaction T is a subset of objects $T \subseteq O$. Thus, the variable m designates the number of objects. Then, an association rule can be defined as an implication:

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

where $X \subset O$, $Y \subset O$, in $X \cap Y = \emptyset$. The following two measures are defined for evaluating the quality of the association rule [1]:

$$conf(X \Rightarrow Y) = \frac{n(X \cup Y)}{n(X)}, \tag{2}$$

$$supp(X \Rightarrow Y) = \frac{n(X \cup Y)}{N}, \tag{3}$$

where $conf(X \Rightarrow Y) \geq C_{min}$ denotes confidence and $supp(X \Rightarrow Y) \geq S_{min}$ support of association rule $X \Rightarrow Y$. There, N in Eq. (3) represents the number of transactions in the transaction database D, and $n(.)$ is the number of repetitions of the particular rule $X \Rightarrow Y$ within D. Additionally, C_{min} denotes minimum confidence and S_{min} minimum support, determining that only those association rules with confidence and support higher than C_{min} and S_{min} are taken into consideration, respectively.

The NARM is very popular concept within the community of researchers who apply the nature-inspired metaheuristics for mining the association rules from transaction databases [7]. The main characteristics of NARM are to handle numerical attribute (i.e., integer and real-valued). Most of the researchers working in this domain apply contemporary EAs as well as SI-based algorithms for solving the problem. All of these algorithms can operate in a larger search space. This situation normally arises in association rule mining, when we deal with larger numbers of attributes and instances in transaction database. Therefore, we can state that these algorithms are very expensive in terms of electrical energy consumption. The stochastic nature of nature-inspired algorithms demanding more runs of an algorithm on a particular dataset increases the time complexity additionally. As a result, the whole ML pipeline has become computationally a very expensive problem.

Interestingly, each numerical attribute in NARM is determined by an interval of feasible values limited by its lower and upper bounds. The broader the interval, the more association rules can be mined. The narrower the interval, the more specific relations between attributes are discovered. Introducing intervals of feasible values has almost two effects on the optimization: To change the existing discrete search space to continuous, and to adapt these continuous intervals to suit the problem of interest better.

3 The Reusable NARM Model

In each run of the nature-inspired algorithm, the algorithm searches for the association rules on the input dataset (sometimes even a raw dataset) which enters the optimization phase. Thus, for each evaluation, the algorithms must perform basic operations, such as for example, calculation of support, confidence or any of the other quality indicators. For most of the datasets which contain numerical attributes, the search space can be almost unlimited. On the one

hand, these operations are very time consuming, while dealing with big datasets demands high computational loads on the other.

These issues were the sources of the main motivation for the development of a new method for building the NARM model, which could consist of two phases (Fig. 1):

- building the model initially,
- reusing the existing model by adding the additional association rules.

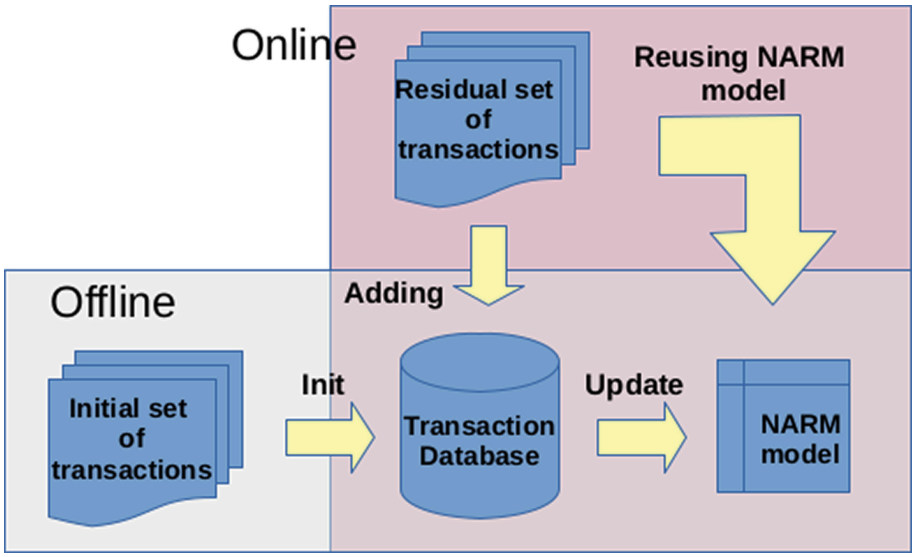

Fig. 1. The concept of reusing NARM model.

The first phase is devoted to the building of a NARM model, and is conducted initially based only on the initial set of transactions. Actually, the model is represented as a set of mined association rules (the gray "offline" rectangle in Fig. 1). In the second phase (the red "online" rectangle in Fig. 1), the NARM model is updated by adding the so-called residual set of transactions that arisen after the initialization step and are saved into the transaction database. Additionally, these transactions represent a basis for modeling the new rule potential capable of reusing the NARM model. Thus, we search for those modelled rules that match the current database the best in the sense of the fitness function. However, the modeled rule can either be added to the reusing NARM model, or ignored in the case that the rule is simply a copy of one already mined. However, the initial model is made by offline NARM method.

The modeling can be defined as an optimization process that can be solved using the nature-inspired population-based algorithms. Each individual, representing the specific modeled rule, is encoded as a real-valued vector:

$$\mathbf{x}_i^{(t)} = \{\underbrace{\langle x_{i,1}^{(t)}, \ldots, x_{i,4}^{(t)}\rangle}_{Attr_1}, \ldots, \underbrace{\langle x_{i,4m}^{(t)}, \ldots, x_{i,4m+3}^{(t)}\rangle}_{Attr_m}, \underbrace{x_{i,4(m+1)}^{(t)}}_{Cp}\}, \tag{4}$$

where each element $x_{i,j}^{(t)}$ for $i = 1, \ldots, Np$ and $j = 1, \ldots, m$ determines a quadruple determining the numerical attribute of a feature $Attr_j^{(t)}$ into the transaction database, D is the number of the attributes, and t is the generation number. Thus, each numerical attribute consists of four real-valued elements decoded as:

$$Attr_{\pi_j}^{(t)} = \begin{cases} x_{i,4j}^{(t)} \mapsto \pi_j^{(t)}, & \text{permutation of feature,} \\ x_{i,4j+1}^{(t)} \mapsto y_{\pi_j}^{(t)}, & \text{lower bound,} \\ x_{i,4j+2}^{(t)} \mapsto z_{\pi_j}^{(t)}, & \text{upper bound,} \\ x_{i,4j+3}^{(t)} \mapsto Th_{\pi_j}^{(t)}, & \text{threshold value,} \end{cases} \tag{5}$$

where permutation $\Pi = (\pi_1, \ldots, \pi_m)$ is needed to modify the ordering of the attribute within the association rules. Technically, all first elements in the corresponding attributes are sorted in descendent order,

$$x_{i,\pi_1}^{(t)} \leq x_{i,\pi_2}^{(t)} \leq \ldots \leq x_{i,\pi_m}^{(t)},$$

while their ordinal values determine their position in the permutation.

The two middle elements encode a real-valued interval of feasible values $[y_{\pi_j}^{(t)}, z_{\pi_j}^{(t)}]$ expressed as:

$$y_{\pi_j}^{(t)} = \begin{cases} \left(Ub_{\pi_j} - Lb_{\pi_j}\right) x_{i,\pi_j+1}^{(t)}, \text{ if } x_{i,\pi_j+1}^{(t)} < x_{i,\pi_j+2}^{(t)}, \\ \left(Ub_{\pi_j} - Lb_{\pi_j}\right) x_{i,\pi_j+2}^{(t)}, \text{ otherwise,} \end{cases}$$

and

$$z_{\pi_j}^{(t)} = \begin{cases} \left(Ub_{\pi_j} - Lb_{\pi_j}\right) x_{i,\pi_j+2}^{(t)}, \text{ if } x_{i,\pi_j+1}^{(t)} < x_{i,\pi_j+2}^{(t)}, \\ \left(Ub_{\pi_j} - Lb_{\pi_j}\right) x_{i,\pi_j+1}^{(t)}, \text{ otherwise.} \end{cases}$$

Interestingly, the integer attributes are treated by the algorithm as real-valued, while their values are truncated when needed.

The threshold value denotes the presence or absence of the π_j-th feature in the observed association rule according to the following equation:

$$Th_{\pi_j}^{(t)} = \begin{cases} enabled, & \text{if } rand(0,1) < x_{i,\pi_j+2}^{(t)} \\ disabled, & \text{otherwise,} \end{cases}$$

where $rand(0,1)$ draws a value from uniform distribution in interval $[0,1]$.

As the last element, the so-called cutting point is added to each vector that distinguishes the antecedent of the rule from the consequent ones. The cutting point Cp is expressed as:

$$Cp_i = \lfloor x_{i,4(m+1)} \cdot (m-1) \rfloor + 1, \tag{6}$$

where $Cp_i \in [1, m-1]$.

The individuals are initialized either heuristically or randomly. When heuristic initialization is used, the lower and upper bounds are determined according to the following equations:

$$\overline{x}_{i,\pi_j} = \frac{o_{\pi_j} - Lb_{\pi_j}}{Ub_{\pi_j} - Lb_{\pi_j}}, \tag{7}$$

where $\overline{x}_{i,\pi_j}$ denotes the encoded value of the numerical attribute, and o_{π_j} is its value in the original problem domain. Then, the corresponding boundary values are expressed as:

$$\begin{aligned} x^{(0)}_{i,\pi_j+1} &= \overline{x}_{i,\pi_j} - \mathcal{N}(\overline{x}_{i,\pi_j}, (Ub_{\pi_j} - Lb_{\pi_j})/2.0), \\ x^{(0)}_{i,\pi_j+2} &= \overline{x}_{i,\pi_j} + \mathcal{N}(\overline{x}_{i,\pi_j}, (Ub_{\pi_j} - Lb_{\pi_j})/2.0), \end{aligned} \tag{8}$$

where $\mathcal{N}(\overline{x}, \sigma)$ is a random number drawn from the Gaussian distribution with mean $\overline{x}$ and standard deviation σ. The motivation about this heuristic initialization is to generate the boundary values of numeric intervals of definite attributes in the vicinity of their original value.

Association rule $X \Rightarrow Y$ is decoded from a representation of an individual, where the quality of the association rule is evaluated using a fitness function. The fitness function is calculated according to the following equation:

$$f\left(\mathbf{x}_i^{(t)}\right) = \frac{\alpha \cdot supp(X \Rightarrow Y) + \beta \cdot conf(X \Rightarrow Y) + \gamma \cdot incl(X \Rightarrow Y)}{\alpha + \beta + \gamma}, \tag{9}$$

where α, β, and γ denote weights, $supp(X \Rightarrow Y)$ and $conf(X \Rightarrow Y)$ represent the support and confidence of the observed association rule, respectively, and $incl(X \Rightarrow Y)$ is defined as follows:

$$incl(X \Rightarrow Y) = \frac{ante(X \Rightarrow Y) + cons(X \Rightarrow Y)}{m}. \tag{10}$$

In Eq. (10), $ante(X \Rightarrow Y)$ 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:

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

The results of the initialization as well as reusing NARM model step are stored into an archive of association rules. As opposed to the traditional algorithms for ARM, this archive is not generated uniformly, where each rule with support and confidence better than S_{min} and C_{min} is stored unconditionally. In our case, only the rules that outperform the best fitness are stored into the archive. In this way, the size of the archive is strongly reduced.

4 Practical Simulations and Evaluations

The purpose of the study was to show that the NARM models can be reused in practice. In line with this, the proposed method was applied to the Wine dataset from the UCI ML repository [4]. Thus, the building of the NARM model step was performed using the 'uARMSolver' framework written in pure C++ [6], while the reusable NARM model initial step was also implemented in C++ by the new 'onlineNARM' algorithm developed according to proposals from the last section. The DE algorithm [9] was applied for realization of both functionalities.

The parameter setting, used during the experimental work is illustrated in Table 1, from which it can be seen that both algorithms applied the same values of parameters except the number of fitness function evaluations. Interestingly, the onlineNARM also employed the heuristic initialization beside the random ones. The random initialization is necessary in order to ensure the diversity of the population. The extensive experiments have shown that the rational bias between heuristic and random initialization was 0.9, i.e. only 10% of randomly generated individuals are needed in the initial population.

Table 1. The DE parameter setting.

Parameter	Symbol	uARMSolver	onlineNARM
Scale factor	F	0.5	0.5
Crossover rate	CR	0.9	0.9
Population size	Np	100	100
Number of fitness function evaluations	$nFEs$	20,000	10,000

The characteristics of the Wine dataset (Table 2) are that it consists of numerical attributes only. This dataset contains data that are obtained as the results of a chemical analysis of wines grown in the same region in Italy, but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines. Additionally, the corresponding class identifier (1–3) is added to the dataset.

Table 2. The characteristics of the Wine dataset.

Characteristic	Original	Reduced	Residual
Attribute characteristics	Numerical	Numerical	Numerical
Number of transactions	178	154	24
Class 1 transactions	59	51	8
Class 2 transactions	71	63	8
Class 3 transactions	48	40	8

However, the original Wine dataset was divided into reduced dataset entering into initialization step, where the NARM model was created initially, and the residual dataset, whose transactions serve for construction of the reusable NARM model in the updating step. Actually, the reduced dataset was obtained by removing 8 of the last incoming transactions in each class.

The results of the proposed method are presented in Table 3, from which it can be seen that the process of reusing the NARM model by onlineNARM achieved the same results as the one of offline construction of the NARM model by uARMSolver .

Table 3. Comparison of two construction methods on the Wine dataset.

The results	uARMSolver		onlineNARM
	Original	Reduced	
Number of association rules	544	524	538
Fitness function value	0.918138	0.917749	0.918138

4.1 Discussion and Issues to Consider

Although the experimental results were obtained according to the Wine dataset only, this dataset is one of the rare ones in the UCI ML repository that comply with the demands of the study, i.e., it consists of the numerical attributes only. The majority of the observed datasets in this repository also support categorical attributes besides the numerical.

The next issue for discussion presents the fact that the observed dataset is slightly small. This comprises of only 178 transactions, which is far from being big data. However, the purpose of this preliminary work was to check if the quality of the results by reusing the NARM model could be comparable with the construction of the NARM model initially, that could be a very time consuming task when big data are observed. The results of the study confirmed this hypothesis.

Interestingly, the NARM model obtained by reducing the original Wine dataset includes the best association rule of less quality than those obtained by searching for the model from the original dataset. This fact confirms that the NARM model was improved by adding the additional association rules of better quality. Last but not least, the time complexity of the onlineNARM is lesser than the uARMSolver due to the looser termination condition. Indeed, only half fitness function evaluations is necessary for the online method onlineNARM to get the comparable results with the original uARMSolver offline method that means even 50% of saving in processing time. This finding complies strongly with the demands of the Green AI.

5 Conclusion

The Green AI refers to those AI methods that are friendly to the environment in the sense of decreased consumption of electrical power. Consequently, the purpose of this study was to develop the Green AI method by introducing the NARM modeling that ensures reusing the NARM models online instead of generating these always anew.

The proposed onlineNARM was applied to Wine dataset member of the UCI ML repository that supports only numerical attributes. The onlineNARM mined the association rules of the similar quality comparing with the results achieved by the offline NARM miner uARMSolver. This was better also in the lower number of the mined association rules and the lesser computational time.

There are a lot of possible directions for the future work. Let us mention only some: (1) to integrate the onlineNARM into a uARMSolver, (2) to apply the proposed onlineNARM to the bigger datasets supporting the numerical attributes, (3) to widen the onlineNARM for applying on datasets of mixed attributes.

Acknowledgements. A. Galvez and A. Iglesias thank the European Union's Horizon 2020 project PDE-GIR under the MSCA-RISE grant agreement No 778035, and the national project #TIN2017-89275-R from the Spanish Ministry of Science, Innovation and Universities (Computer Science National Program), Agencia Estatal de Investigación and European Funds FEDER (AEI/FEDER, UE). I. Fister and I. Fister Jr. thank the financial support from the Slovenian Research Agency (Re- search Core Funding No. P2-0042 and No. P2-0057, respectively).

References

1. Agrawal, R., Srikant, R., et al.: Fast algorithms for mining association rules. In: Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, vol. 1215, pp. 487–499 (1994)
2. Blum, C., Merkle, D.: Swarm Intelligence: Introduction and Applications. Springer Publishing Company, Incorporated, 1 edition. (2008). https://doi.org/10.1007/978-3-540-74089-6_2
3. Dhar, P.: The carbon impact of artificial intelligence. Nat. Mach. Intell. **2**, 423–5 (2020)
4. Dua, D., Graff, C.: UCI machine learning repository (2017)
5. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer Publishing Company, Incorporated, 2nd edition (2015). https://doi.org/10.1007/978-3-662-05094-1
6. Fister, I., Fister Jr. I.: uARMSolver: a framework for association rule mining. CoRR, arXiv:abs/2010.10884 (2020)
7. Fister Jr, I., Fister, I.: A brief overview of swarm intelligence-based algorithms for numerical association rule mining. arXiv preprint arXiv:2010.15524 (2020)
8. Schwartz, R., Dodge, J., Smith, N.A., Etzioni, O.: Green AI. arXiv preprint arXiv:1907.10597 (2019)
9. Storn, R., Price, K.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**(4), 341–359 (1997)
10. Strubell, E., Ganesh, A., McCallum, A.: Energy and policy considerations for deep learning in NLP. arXiv preprint arXiv:1906.02243 (2019)