



A Systematic Assessment of Numerical Association Rule Mining Methods

Minakshi Kaushik¹ · Rahul Sharma¹ · Sijo Arakkal Peious¹ · Mahtab Shahin¹ · Sadok Ben Yahia² · Dirk Draheim¹

Received: 18 March 2021 / Accepted: 22 May 2021 / Published online: 22 June 2021
© The Author(s), under exclusive licence to Springer Nature Singapore Pte Ltd 2021

Abstract

In data mining, the classical association rule mining techniques deal with binary attributes; however, real-world data have a variety of attributes (numerical, categorical, Boolean). To deal with the variety of data attributes, the classical association rule mining technique was extended to numerical association rule mining. Initially, the concept of numerical association rule mining started with the discretization method, and later, many other methods, e.g., optimization, distribution are proposed in state-of-the-art. Different authors have presented various algorithms for each numerical association rule mining method; therefore, it is hard to select a suitable algorithm for a numerical association rule mining task. In this article, we present a systematic assessment of various numerical association rule mining methods and we provide a meta-study of thirty numerical association rule mining algorithms. We investigate how far the discretization techniques have been used in the numerical association rule mining methods.

Keywords Knowledge discovery in databases · Data mining · Association rule mining · Numerical association rule mining · Quantitative association rule mining

This work has been partially conducted in the project “ICT programme” which was supported by the European Union through the European Social Fund.

This article is part of the topical collection “Future Data and Security Engineering 2020” guest edited by Tran Khanh Dang.

✉ Minakshi Kaushik
minakshi.kaushik@taltech.ee

Rahul Sharma
rahul.sharma@taltech.ee

Sijo Arakkal Peious
sijo.arakkal@taltech.ee

Mahtab Shahin
mahtab.shahin@taltech.ee

Sadok Ben Yahia
sadok.ben@taltech.ee

Dirk Draheim
dirk.draheim@taltech.ee

¹ Information Systems Group, Tallinn University of Technology, Akadeemia tee 15a, 12618 Tallinn, Estonia

² Software Science Department, Tallinn University of Technology, Akadeemia tee 15a, 12618 Tallinn, Estonia

Introduction

In today’s scenario, data is growing explosively, and it is available in many various forms (numerical, text, images, etc.). To manage this humanly unmanageable large amount of data, researchers and data scientists have developed many techniques. In knowledge discovery in databases (KDD), data mining is a popular technique for extracting the required information and finding patterns between data items. Association rule mining (ARM), classification, clustering, regression, etc., are a few well-known data mining techniques. Agrawal [2] introduced ARM in 1993 for finding the relationship between different data items, and later, he proposed the Apriori [3] algorithm and its version to discover interesting rules in large databases. ARM is widely used in market basket analysis, medical diagnosis, and bioinformatics. Apriori and FP-growth [28] are also the most popular algorithms in classical association rule mining. Different authors have various opinions about the discretization process and ARM. Recently, Draheim [18] “provides a frequentist semantics for conditionalization on partially known events, which is given as a straightforward generalization of classical conditional probability via so-called probability testbeds.”

The classical association rule mining deals only with the binary attributes, whereas real-world data have mixed attributes (numerical, categorical). Therefore, whenever data is in numerical form (height, weight, or age), the data items need to be changed from numerical to discrete using a discretization process. This process of finding association rules in numerical data items has been referred to as numerical association rule mining (NARM) or quantitative association rule mining (QARM) [60]. Initially, NARM was started with the discretization method, and later many methods (optimization, discretization, distribution) are proposed in the literature. Therefore, many other authors investigated the discretization method and proposed various alternatives to the discretization method.

In the literature, various methods with multiple algorithms are discussed; however, selecting an appropriate algorithm for a NARM task with valid reasons is not yet discussed. This article extends our previous work [32] and provide a detailed study of thirty NARM algorithms under different NARM methods. We also investigate how far the discretization techniques have been used in the numerical association rule mining methods.

We conduct an automated search process over Scopus Database and manual search on Google Scholar. We decide to have the term (“Numerical Association Rule Mining” OR “Quantitative Association Rule Mining”) to search in abstract, title, and keyword. Our research is limited to the articles published between the years 1996-2020. The selected papers are again assessed on the following criteria:

- Papers introducing novel algorithm in numerical association rule mining or quantitative association rule mining.
- Papers extending the existing algorithm in numerical association rule mining or quantitative association rule mining.

Moreover, we use the following criteria to exclude the papers from the list of searched papers:

- Papers introducing the application of NARM algorithm in any field.
- Papers published in languages other than English.
- Technical reports, thesis and other documents had no peer-review process.

The paper is structured as follows. In section “[Preliminaries](#),” we describe preliminaries. In section “[Methods to Solve Numerical ARM Problems](#),” we discuss all three methods to solve numerical association rule mining problems. In section “[The Optimization Method](#),” the optimization method is discussed with all its sub-methods. In section “[The Distribution Method](#),” the distribution method is introduced and discussed, and in section “[The Discretization Method](#),” the

discretization method is discussed. A discussion on various methods and algorithms is given in section “[Discussion](#).” The conclusion is given in section “[Conclusion](#).”

Preliminaries

In this section, we provide basic introductions about ARM and NARM.

Association rule mining

In ARM, association rules are based on the If-then relations, which consist of antecedents (If) and consequents (Then) [2]. For example, (1) shows the following association rule: “If a customer buys bread, then he also buys milk.” Here, Bread appears as antecedent and Milk as consequent. Generally, an association rule may be represented as a production rule in an expert system, an *if statement* in a programming language, or an implication in a logical calculus.

$$\{\text{Bread}\} \Rightarrow \{\text{Milk}\} \quad (1)$$

In a database, let I be a set of m binary attributes $\{i_1, i_2, i_3, \dots, i_m\}$ called database items. Let T be a set of n transactions $\{t_1, t_2, t_3, \dots, t_n\}$, where each transaction t_i has a unique ID and consists of a subset of the items in I , i.e., $t_i \subseteq I$. As in (1), an association rule is an implication of the form

$$X \Rightarrow Y \quad (2)$$

where $X, Y \subseteq I$ (itemsets) and $X \cap Y = \emptyset$. An association rule can be extracted on the basis of two important measures: support and confidence. Support of an association rule can be defined as the percentage of transactions of the total records containing both sets of items X and Y that are $(X \cup Y)$. Confidence of an association rule can be described as the percentage of transactions that contain X also contain Y .

$$\text{Support}(X \Rightarrow Y) = \text{Supp}(X \cup Y) \quad (3)$$

$$\text{Confidence}(X \Rightarrow Y) = \frac{\text{Supp}(X \cup Y)}{\text{Supp}(X)} \quad (4)$$

For instance, with the reference of Table 1, we can understand the concept of support and confidence. The support of the association rule (Bread \Rightarrow Milk) is $2/6 = 0.33$. Since both items are bought together two times out of six transactions, so support is called 20%. However, both items are bought together two times out of four transactions that contain Bread. This indicates the confidence $2/4 = 0.5$ is 50%.

In ARM, to find out the interesting rules, various interestingness measures are proposed in the literature [58]. In

Table 1 Market basket analysis in association rule mining

TID	Items
T1	Milk, butter
T2	Butter, bread
T3	Sugar, milk
T4	Milk, bread
T5	Sugar, bread
T6	Milk, sugar, bread

classical ARM, frequent itemsets and association rules are discovered from a Boolean dataset; therefore, it is also known as binary or Boolean ARM. Table 2 shows a Boolean dataset for classical ARM. This table contains attributes corresponding to each item and a row corresponding to each transaction. Each attribute has a value “1” if the item is available in the transaction else “0”.

Numerical Association Rule Mining

To extract association rules from numerical data, the problem of the quantitative or categorical attribute was first discussed by Srikant in 1996 [60]. In NARM, whenever data is in numerical form (height, weight, or age), the data items need to be changed from numerical to discrete using a discretization process. This process of finding association rules in numerical data items has been referred to as numerical association rule mining (NARM) [60]. NARM can easily be understood by the following example.

Age [25, 40] \wedge Gender : [Female] \Rightarrow Salary [1300, 2000]
(Supp = 30%, Confidence = 60%)

Given a set of transactions T , let Antecedent denote the set of transactions in T in which Age has a value between 25 and 40 and Gender is Female. Similarly, let Consequent denote the set of transactions in which Salary has a value between \$1300 and \$2000. For instance, with reference to Table 3, here Supp = 30% denotes that 30% of the employees are females and between the ages 25 and 40, earning a salary of between \$1300 and \$2000. Conf = 60% denotes that 60% of the female employees between age 25 and 40 are

Table 2 Example of Boolean dataset

TID	Milk	Butter	Bread	Sugar
T1	1	1	0	0
T2	0	1	1	0
T3	1	0	0	1
T4	1	0	1	0
T5	0	0	1	1
T6	1	0	1	1

Table 3 Example of numerical values dataset

Age	Gender	Salary
25	Male	1200
26	Female	1250
28	Female	1250
30	Female	1350
35	Female	1600
38	Female	1700
40	Male	1900
42	Male	1950
48	Female	2500
50	Male	3000

earning a salary of between \$1300 and \$2000. Here Age and Salary are numerical attributes and Gender is a categorical attribute.

As an early solution, the problem of association rules for numerical data was solved using a discretization process where numeric attributes are divided into different intervals and, henceforth, these attributes are treated as categorical attributes [12]. For example, an attribute Age with values between 20 and 80 can be divided into six different age intervals (20–30, 30–40, 40–50, 50–60, 60–70, 70–80). The data discretization process is an obvious solution; however, it reveals a loss of valuable information, which might cause poor results [17]. Thus, we review solutions from three different approaches (discretization, distribution and optimization) to solve issues with numerical association rule mining in section “Methods to Solve Numerical ARM Problems.”

Methods to Solve Numerical ARM Problems

To solve the issues in NARM, three main approaches (discretization, distribution and optimization) have been discussed in the literature. Based on these three approaches, many different NARM algorithms are proposed. The optimization method has several sub-methods as swarm intelligence and evolution-based algorithms, covering most of the area to deal with NARM. The distribution method does not contribute much in this area; however, the discretization method is a common method that transforms continuous attributes into discrete attributes and it is further subdivided into three sub-methods. Figure 1 (also compared with Fig. 1 in [9]) shows all three approaches and different algorithms proposed under each approach.

The Optimization Method

To solve the NARM problems, many researchers have moved towards optimization methods. Optimization methods provide a robust and efficient approach to explore a massive

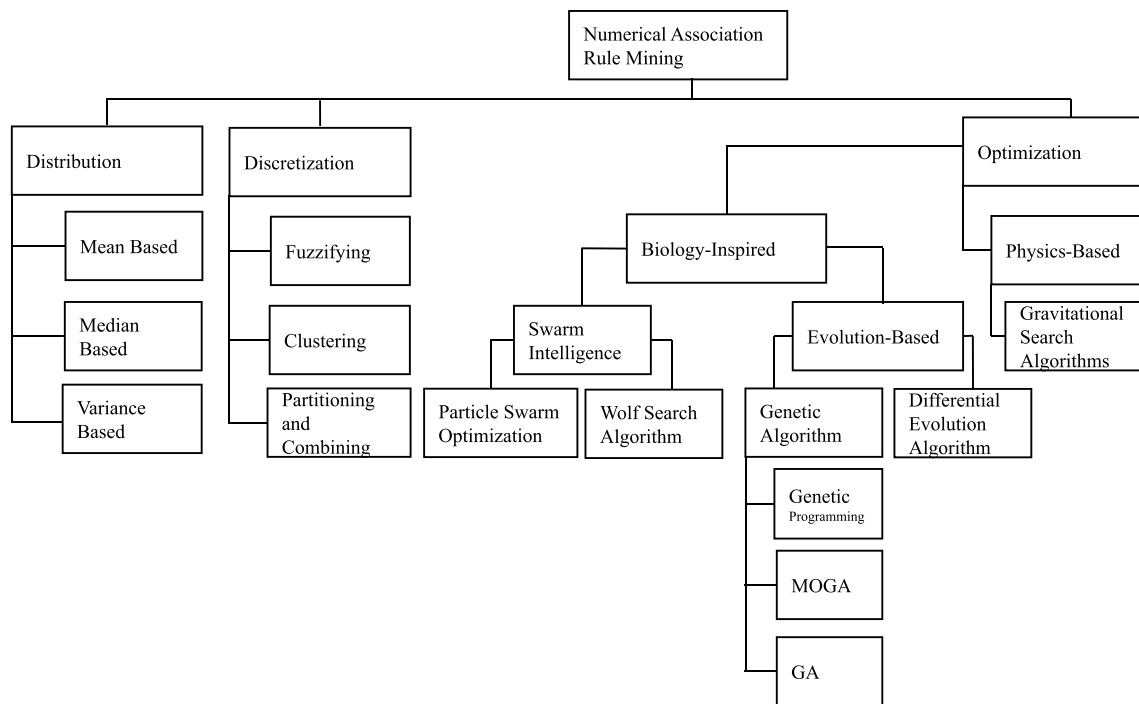


Fig. 1 Different methods and algorithms to solve numerical association rule mining problems

search space. In this method, researchers have invented a collection of heuristic optimization methods inspired by the movements of animals and insects. For finding association rules, optimization methods work in two phases. In the first phase, all the frequent itemsets are found and in the second phase, all relevant association rules are extracted. As shown in Fig. 1, optimization methods are divided into bio-inspired optimization and physics-based optimization methods. Table 4 shows an overview of all those algorithms that come under the optimization method.

The Bio-inspired Optimization Method

Biology-based algorithms are generally divided into two parts: swarm-intelligence-based algorithms and

evolution-based algorithms [15]. The main origin of these algorithms is the biological behavior of natural objects [68].

Evolution-Based Algorithms

Evolution-based algorithms are inspired by Darwinian principles and were first applied in [48]. These algorithms mimic the capability of nature to develop living beings that are well-adapted to their environment [68]. Evolution-based algorithms exploit stochastic search methods that follow the idea of natural selection and genetics. The algorithms show strong adaptability and self-organization [15] and use biology-inspired operators such as crossover, mutation, and natural selection [68]. The *Genetic Algorithm* [30] and the *Differential Evolution Algorithm* [63]

Table 4 An overview of optimization method algorithms for NARM

Methods	Basic technique	Algorithms
GA	Genetic Algorithm	GENAR [47], GAR [48], EGAR [39], ARMGA [74], EARMGA [74], GAR-PLUS [10], QuantMiner [56], RelQM-J [57], RCGA [46]
MOGA	Genetic Algorithm	ARMMGA [53], QAR-CIP-NSGA-II [45]
DE	Differential Evolution	MODENAR [7], ARM-DE [20]
PSO	Particle Swarm Optimization	RPSO [4], CENPSO [5], MOPAR [12], PPQAR [73], PARCD [65]
WSA	Swarm Intelligence	WSA [1]
GSA	Physics-based (Gravity)	GSA [13]

are two examples of evolution-based algorithms. Table 5 shows an overview of the evolution-based algorithms for NARM, together with concepts.

Genetic Algorithms (GA) GA was first proposed by Holland [30] and they are one of the most popular algorithms in bio-inspired optimization methods. A basic genetic algorithm consists of five phases: initialization, evaluation, reproduction, crossover, and mutation. GAs for NARM can be divided into three fields, i.e., basic genetic algorithms, genetic programming and multiobjective genetic algorithms. A basic genetic algorithm has been proposed by Mata et al. [47] and together with the tool GENAR (GENetic Association Rules) to discover association rules with numeric attributes. With this tool, an undetermined amount of numeric attributes in antecedent and unique numeric attribute in consequent can be obtained. Association rules in GENAR algorithms allow for intervals (maximum and minimum values) for each numeric attribute. Mata et al. [48] further extended the GENAR algorithms and proposed a technique named GAR (Genetic Association Rule) to discover association rules in numeric databases without discretization. Authors present a technique to find frequent itemsets in numeric databases without needing to discretize numeric attributes. This algorithm was useful only for finding the frequent itemsets, not for association rules. In this paper, a genetic algorithm was used to find the suitable amplitude of the intervals that conform k -itemset and can have a high support value without too wide intervals. In [39], the GAR algorithm was further extended to EGAR (extended genetic association

rule). This algorithm generates frequent patterns with continuous data [48].

A genetic-based strategy and two other algorithms ARMGA and EARMGA, were proposed by Yan et al. [74]. In this approach, an encoding method was developed with relative confidence as the fitness function. ARMGA was proposed for Boolean ARM and EARMGA for quantitative attributes or generalized association rules. In these algorithms, there was no requirement of a minimum support threshold. The GAR-plus tool was presented by Alvarez [10]. This tool deals with categorical and numeric attributes in large databases without any need for a prior discretization of numeric attributes.

In 2013, Salleb et al. [56] proposed “Qu antMiner, a quantitative association rule mining system based on the genetic algorithm. This tool dynamically discovers meaningful intervals in association rules by optimizing both the confidence and the support values.

Seki and Nagao [57] worked on GA-based QuantMiner for multi-relational data mining and developed RelQM-J, a tool for relational quantitative association rules in Java programming language. In this tool, efficient computation of the support of the rules has been realized by using a hash-based data structure.

A real-coded [36] genetic algorithm was presented in [46] in 2010. The proposed algorithm RCGA follows the CHC binary-coded evolutionary algorithm [19]. RCGA algorithm has been applied to pollutant agent time series and helps to find all existing relations between atmospheric pollution and climatological conditions.

Table 5 An overview of evolution based algorithms for NARM

Algorithm	Proposer	Concept
GENAR [47]	J. Mata Vázquez et al. (2001)	Based on finding frequent itemsets in numerical databases and intervals of all attributes that conform to those frequent itemsets
GAR [48]	J. Mata Vázquez et al. (2002)	Extended version of GENAR
EGAR [39]	H. Kwaśnicka et al. (2006)	Uses medical databases where attributes are continuous and discrete; extended version of GAR.
EARMGA [74]	A. Yan et al. (2009)	No requirement of minimum support threshold.
GARPLUS [10]	V. Álvarez et al. (2012)	Based on the finding intervals of numeric attribute.
QUANTMINER [56]	A. Salleb Aouissi et al. (2013)	Based on genetic algorithm to find good intervals by optimizing both support and confidence.
RelQM-J [57]	H. Seki (2017)	Based on mining numeric rules from relational databases, implemented in Java
RCGA [46]	M. Martínez Ballesterosa (2010)	Based on CHC binary-coded evolutionary algorithm.
ARMMGA [53]	H. Reza Qodmanan (2011)	Based on multi-objective genetic algorithm.
QAR-CIP-NSGA-II [45]	D. Martín et al. (2011)	Based on NSGA with three measures (comprehensibility, interestingness, performance).
MODENAR [7]	B. Alatas (2008)	Based on multi-objective differential evolutionary algorithm.
ARM-DE [20]	I. Fister Jr. (2018)	Single objective optimization problem where features consist of numerical as well as categorical attributes.

Genetic Programming for ARM Genetic Programming [37] is a well-known type of GA. In GA, the genome is in string structure, while in GP, the genome is in the form of tree structure [29]. Genetic Network Programming (GNP) is a graph-based evolutionary algorithm and finds the association rules for continuous attributes. In this method, important rules are stored in a pool and these extracted rules are measured by the chi-squared test. This pool is updated in every generation by exchanging the association rule with a higher chi-squared value for the same association rule with a lower chi-squared value [64].

Multi-Objective Genetic Algorithm The multi-objective genetic algorithm was proposed by Fonseca et al. [21] in 1993. Generally, the resource consumption of an association rule mining computation is affected by two parameters, i.e., minimum support and minimum confidence. In classical ARM algorithms, only a single measure (support or confidence) has been used as a measure to evaluate the rule interestingness, therefore, if the values of minimum support and minimum confidence are not appropriately set, then the number of association rules may be significantly less, or it may be very large. This problem can be solved by using more objectives or measures as referred to in multi-objective ARM.

Gosh and Nath [23] used a Pareto-based genetic algorithm to solve the multi-objective rule mining problem using three measures: interestingness, comprehensibility and predictive accuracy. The single-objective algorithm, ARMGA [74], had issues that were addressed by introducing the multi-objective genetic algorithm called ARMMGA by Qodmanan et al. in [53]. The ARMGA algorithm finds high confidence and low support rules, whereas ARMMGA finds high confidence and high support rules. ARMGA has a large set of rules compared to ARMMGA; this problem was solved using a new fitness function in ARMMGA. To prevent invalid chromosomes in ARMGA, new crossover and mutation operators are presented in the literature.

Srinivasan and Deb [61] proposed a non-dominated genetic sorting algorithm to solve multi-objective optimization problems. In 2002, Deb et al. [16] extended NSGA to NSGA-II. In 2011, Martin et al. [45] extended NSGA-II with a trade-off between interpretability and accuracy. NSGA-II performs evolutionary learning of intervals of attributes. For each rule, condition selection is made for three objectives (interestingness, comprehensibility and performance). This method did not depend on minimum support and confidence thresholds. Martin et al. again extended their research on NSGA-II to a new approach called QAR-CIP-NSGA-II and compared the results of this algorithm with other MOEA(Multi-objective evolutionary algorithm) algorithms.

Differential Evolutionary Algorithms Differential evolutionary (DE) algorithms are evolution-based algorithms. These algorithms were proposed by Storn and Price in [62]. DE algorithms are simple and effective single-objective optimization algorithms that solve real-valued problems based on the principle of natural evolution. DE algorithms use Genetic-based operators such as crossover, mutation, and selection. Although the evolution process of DE is similar to the one of GA, it relies on a mutation operator instead of a crossover operator [69].

A Pareto-based multi-objective DE algorithm for ARM was first proposed in [7] by Alatas et al. for searching accurate and comprehensible association rules. The problem of mining association rules was formulated with four objective optimization problems, i.e., support, confidence, comprehensibility and amplitude. Here, support, confidence and comprehensibility are maximization objectives and the amplitude of intervals is a minimization objective. In a single run, a Pareto-based multi-objective DE algorithm search intervals of numeric attributes and association rules.

In 2018, [20] proposed a novel approach for mining association rules with numerical and categorical attributes based on DE. In this algorithm, a single objective optimization problem is considered in which support and confidence of association rules are combined into a fitness function. This new DE using ARM (ARM-DE) with mixed (i.e., numerical and categorical) attributes consists of three stages: (1) domain analysis, (2) representation of a solution, (3) definition of a fitness function.

Swarm Intelligence Based Algorithms

Swarm intelligence-based algorithms are further divided into two sub-optimization methods, particle swarm optimization and the wolf search algorithm. Table 6 provides an overview of swarm intelligence algorithms for NARM.

Particle Swarm Optimization Particle Swarm Optimization (PSO) is a population-based optimization algorithm for nonlinear functions. This algorithm is oriented towards animal behavior, such as bird flocking or fish schooling. It was developed in 1995 [33, 52]. PSO was first used for NARM to find intervals of the numerical attributes in 2008 [4].

Rough PSO, based on rough patterns, was proposed in [4], in which rough values are defined with upper and lower intervals. This algorithm can complement the existing tools developed in rough computing. Rough values are helpful in representing an interval for an attribute. In this work, each particle consists of a decision variable that has three parts. The first part of each decision variable represents the antecedent or consequent of the rule and can take values between 0 and 1. The second part describes the lower bound; the third part represents the upper bound of the item interval.

Table 6 An overview of Swarm intelligence based algorithms for NARM.

Algorithm	Proposer	Concept
RPSO [4]	B. Alatas et al. (2008)	RPSOA is based on the notion of rough patterns that use rough values defined with upper and lower intervals.
CENPSO [5]	B. Alatas et al. (2009)	CENPSO is based on chaos numbers.
MOPAR [12]	V. Beiranvand et al. (2014)	MOPAR is Based on Multi objectives (confidence, comprehensibility and interestingness).
MOPSO [38]	Kuo et al. (2019)	Based on pareto optimal strategy using adaptive archive grid for multi-objective PSO.
Parallel PSO [73]	A. Yan et al. (2019)	Parallel PSO is based on two methods of parallel algorithm: particle-oriented and data-oriented parallelization.
PARCD [65]	I. Tahyudin et al. (2017)	Combined PSO method with cauchy distribution.
WSA [1]	I.E. Agbehadji et al. (2016)	Based on wolves hunting strategy.

The second and third parts are combined as one rough value during the implementation phase of particle representation.

Alatas and Akin [5] proposed a novel PSO algorithm based on chaos numbers. The CENPSOA algorithm (chaotically encoded PSO) uses chaos decision variables and chaos particles. Chaos and PSO relation were first discovered by Liu et al. [42]; the CENPSOA algorithm performs encoding of particles given by chaos numbers. The Chaos numbers consist of the midpoint and radius part of values [5]. Alatas and Akin [6] also proposed a multi-objective chaotic particle swarm optimization algorithm for mining accurate and comprehensible classification rules.

Yan et al. [73] proposed a parallel PSO algorithm for NARM. This parallel algorithm was designed with two strategies called particle-oriented and data-oriented parallelization. Particle-oriented parallelization is more efficient and data-oriented parallelization is more scalable to process large datasets.

To discover association rules in a single step without prior discretization of numerical attributes, Beiranvand et al. [12] proposed a multi-objective particle swarm optimization algorithm (MOPAR). The algorithm defines multiple objectives such as confidence, comprehensibility and interestingness. In the Pareto method, a candidate solution is identified better than all other candidates. In multi-objective optimization, a set of best solutions is identified in which the members are superior among all the candidates.

Kuo et al. [38] proposed a multi-objective particle swarm optimization algorithm using an adaptive archive grid for NARM. It is also based on Pareto's optimal strategy. In this algorithm, minimum support and minimum confidence are not required before mining. MOPSO algorithm is executed in three parts: (1) initialization, (2) adaptive archive grid, and (3) particle swarm optimization searching.

PSO for NARM with Cauchy distribution (PARCD) has been evaluated by [65] and it showed that the result of PARCD is better than the method of MOPAR.

Wolf Search Algorithm The wolf search algorithm (WSA) is a bio-inspired heuristic optimization algorithm. It was proposed by [67] and imitated the way wolves search for food and survive by avoiding their enemies. WSA is tested and compared with other heuristic algorithms and investigated with respect to its memory requirements. The group of wolves has characteristics of commuting together as a nuclear family; that is why it is different from particle swarm optimization [72].

Agbehadji and Fong [1] proposed a new meta-heuristic algorithm that used the wolf search algorithm for NARM. The wolf has three different features of preying. These are prey initiatively, prey passively and escape. The preying initiatively feature allows the wolf to check its visual perimeter to detect prey. If the prey is found within visual distance, the wolf moves towards the prey with the highest fitness value; else, the wolves will maintain their direction. In prey passively mode, the wolf only stays alert from threats and tries to improve its position. In the escape mode, when a threat is detected, the wolf escapes quickly by relocating itself to a new position with an escape distance greater than its visual range.

Physics-Based Algorithm

The physics-based meta-heuristic optimization algorithm simulates the physical behavior and properties of the matter or follows the laws of physics [15]. For NARM, the gravitational search algorithm is a physics-based meta-heuristic optimization algorithm.

Gravitational Search Algorithm

Rashedi et al. proposed a new optimization algorithm based on the law of gravity and named it gravitational search algorithm (GSA) [54]. Newtonian gravity laws state that "Every particle in the universe attracts every other particle with a force that is directly proportional to the product of their masses and inversely proportional to the square of

the distance between them.” In GSA, agents act as objects and their performance is evaluated by their mass. Each mass presents a solution and it is expected that masses will be attracted by the heaviest mass. GSA is like a small artificial world of masses obeying the Newtonian laws of gravitation and motion. There are four ways of representing the agents or coding the problem variables. These are continuous (real-valued), binary-valued, discrete, and mixed, which are called GSA variants [55].

Can and Alatas [13] first used GSA for NARM. GSA eliminated the task of finding the minimum values of support and confidence. Automatically mined rules have high confidence and support values. In this work, GSA has been designed to automatically find the numerical intervals of the attributes, i.e., without any *a priori* data process at the time of rule mining. The problem of interactions within attributes has been eliminated with the designed GSA by not selecting one attribute at a time and not evaluating a partially-constructed candidate rule due to its global searching with a population.

The Distribution Method

In [11], Aumann and Lindell have introduced a new definition for numerical association rules based on statistical inference theory. In this study, they have implemented several distribution scales, including mean, median, and variance. The following example shows the kind of generalization of ARM proposed by the authors.

$$\text{Gender} = F \Rightarrow \text{Wage} : \text{mean} = \$8.50$$

$$(\text{overall mean wage} = \$12.60) \quad (5)$$

As the above example shows, the average wage for females was \$ 8.50 p/hr. The rule displays that the wage of that group was far less than the average wage; therefore, this rule can be considered useful. They also used the algorithm to identify repeated itemsets and then calculate the desired statistics for the purpose with respect to repeated itemsets. This procedure is restricted by the requirement to store every repeated itemsets in memory throughout repeated itemset generation. Where the data is not sparse, the number of frequent itemsets will be huge and repeated itemset storage and access will dominate the calculation. Moreover, they concluded that the suggested algorithm is beneficial and may find rules between two given quantitative attributes. Webb [71] extended the work proposed by Aumann and Lindell in [11] with name impact rules using the OPUS search algorithm [70]. In this paper, the author evaluated the impact of conditions on a numeric variable that association rules with discretization can not emulate. The author compared the frequent itemset approach with the OPUS_IR approach. The author found

OPUS_IR avoids large memory requirements with a frequent itemset approach by avoiding the need to store all frequent itemsets.

The Discretization Method

Discretization is a process of quantizing numerical attributes into groups of intervals and it is one of the most popular methods to solve the problem of numerical association rule mining. There are numerous methods of discretization in literature. Due to different needs, discretization methods have been developed in different ways, such as supervised vs. unsupervised, dynamic vs. static, global vs. local, splitting (top-down) vs. merging (bottom-up) and direct vs. incremental [43]. In classical ARM algorithms, numerical columns cannot be processed directly [44], i.e., all columns need to be categorical, which is a major limitation of ARM [66].

Discretization of numerical values is used to overcome this problem [34, 49, 50]. When a numeric column is divided into useful target groups, it becomes easier to identify and generate association rules, i.e., discretization helps to understand the numeric columns better. The discretized groups are useful only if the variables in the same group do not have any objective difference. Discretization minimizes the impact of trivial variations between values. Discretization can be performed using fuzzifying, clustering and partitioning and combining [8]. In Table 7, we summarize some selected discretization algorithms used in NARM.

Fuzzifying

Fuzzifying is the technique of illustrating numeric values as fuzzy sets [35], which can help to rectify the *sharp boundary problem* of ARM. Sometimes, endpoint values of discretized groups have more or less influence on the result than the midpoint values: this phenomenon is known as a sharp boundary problem. Fuzzy Class Association Rule Support Vector Machine (FCARSVM) is a model proposed by Kianmehr et al. [35] to get the fuzzy class association rules. In the first phase of the model, Fuzzy class association rules (FCAR) are extracted using fuzzy c-means clustering algorithm for quantitative datasets and in the second phase, extracted FCARs are weighted based on scoring metric strategy.

For mining fuzzy quantitative association rules, those have crisp values, fuzzy terms and intervals in both antecedent and consequent, Zhang [76] presented an algorithm EDPFT(equal-depth partition with the fuzzy term). The author used an equal-depth partition algorithm for finding the intervals of numeric values and map crisp values and fuzzy terms of each categorical attribute into consecutive integers and generate frequent itemsets using

Table 7 An overview of discretization-based algorithms for NARM

Algorithm	Proposer	Concept
ARCS [40]	B. Lent et al. (1997)	Based on segmenting clusters using the geometric-based BitOp algorithm.
DRMiner [41]	W. Lian (2005)	Based on finding density regions in a multidimensional space.
DBSMiner [25]	G. Yunkai et al. (2008)	Based on clustering of high density sub-spaces using a density- and grid-based cluster algorithm.
MQAR [75]	Y. Junrui et al. (2010)	Based on finding dense sub-spaces using structure DGFP-tree.
Srikant's Partitioning and combining algorithm [60]	R. Srikant (1996)	Partitioning the numeric attribute into interval using equi-depth method
APACS2 [14]	K.C.C. Chan et al. (1997)	Based on partitioning approach
EDPFT [76]	W. Zhang (1999)	Based on fuzzifying approach with equal-depth partition method
FTDA [31]	Hong et al. (1999)	Based on fuzzifying approach with apriori algorithm
OFARM [77]	H. Zheng et al. (2014)	Based on fuzzifying approach with multi-objective functions

the extended apriori algorithm. In 1999 Hong et al. [31] also proposed an algorithm FTDA (fuzzy transaction data-mining algorithm), which integrates the fuzzy-set concepts with an apriori algorithm. This method encounters the problem of requiring the fuzzy-sets and their corresponding membership functions in advance. Choosing the best fuzzy-sets for mining the association rule is difficult, as anomalies may occur if fuzzy-sets are not well chosen. To tackle this problem, [26] introduced an additional fuzzy normalization process and proposed an algorithm for fuzzy quantitative association rules. [26] also compared with normalization and without normalization methods for mining fuzzy quantitative rules and show with normalization method gives a high number of interesting rules compare to with normalization method. The authors used three interest measures: fuzzy support, fuzzy confidence, and fuzzy correlation. In 2014, [77] proposed a novel algorithm OFARM (optimized fuzzy association rule mining) to optimize the partition points of fuzzy sets with multiple objective functions. A two-level iteration process is used to generate the frequent itemsets and employ certainty factor with confidence to evaluate fuzzy association rules.

Clustering

Clustering is one of the popular methods of discretizing a numerical column in an unsupervised manner [8]. In clustering, a numerical column is segregated into different groups according to the properties of each value; in this method, the probability of having values in the same group depends on the degree of similarity or dissimilarity of the values [27, 59]. To obtain maximum results in clustering, the degree of similarity and dissimilarity needs to be well defined [24]: “In other words, the intra-cluster variance is to be minimized, and the inter-cluster variance is to

be maximized” [66]. Two-step clustering [59] is the most common clustering method.

DRMiner Algorithm

Lian et al. [41] have proposed the DRMiner algorithm, which exploits the notion of “density” to capture the characteristics of numeric attributes and an efficient procedure to locate the “dense regions.” DRMiner scales up well with high-dimensional datasets. When mapping a database to a multi-dimensional space, the data points (transactions) are not distributed evenly throughout the multi-dimensional space. For this kind of distribution, the density measure was introduced and the problem of mining quantitative association rules transformed into the problem of finding dense regions to map them to find quantitative association rules. Weaknesses of this method were the prior requirement of many thresholds and unsolving the dimensionality curse. It was noted that the algorithm might not perform well for datasets with uniform density between minimum density threshold and low density.

DBSMiner

DBSMiner is a density-based sub-space mining algorithm using the notion of density-connected to cluster the high-density sub-space of numeric attributes and gravitation between grid/cluster to deal with the low-density cells [25]. DBSMiner employs an efficient high dimension clustering algorithm CBSD (Clustering Based on Sorted Dense unit) to deal with high dimensional data sets. The algorithm has a unique feature to deal with low-density sub-spaces and there is no need to scan the whole space; check the neighbor cell. It can find interesting association rules.

MQAR

MQAR (Mining Quantitative Association Rules based on a dense grid) is a novel algorithm that was proposed by Yang and Zhang [75]. The main objective of this algorithm was to mine the numeric association rules using a tree structure, DGFP-tree, to cluster dense space. This algorithm is helpful to eliminate noise and redundant rules by transforming the problem into finding regions with enough density and to map them to quantitative association rules. A novel subspace clustering algorithm was also proposed based on searching DGFP-tree and inserting the dense cell in the database space into DGFP-tree as a path from a root node to a leaf node. MQAR has the advantage that DGFP-tree compresses the database and there is no need to scan the database several times.

ARCS

The Association Rule Clustering System [40] was presented by Lent et al. together with a new geometric-based clustering algorithm, BitOP. In this paper, the problem of clustering of association rules like $(A \wedge B) \Rightarrow C$ where L.H.S. is having quantitative attributes and R.H.S. having a categorical attribute was discussed and a two-dimensional grid is formed where each axis represents one of the L.H.S. attributes. ARCS is an automated system to compute a clustering of two-attribute spaces in large databases. In ARCS framework Binner, For a given partitioning of the input attributes, the algorithm makes only one pass through the data. and allows the support or confidence thresholds to change without

requiring a new pass through the data. BitOp algorithm enumerates the clusters. To locate clusters within bitmap grids, the algorithm performs bit-wise operations.

Partitioning and Combining

In [60], Srikant and Agrawal discussed the problems of numeric attributes in databases. The authors addressed the issue of mining association rules from large databases containing both numerical and categorical attributes. A partitioning method was introduced to deal with this problem, which partitions quantitative attributes into intervals and map pairs (attribute, interval) to Boolean attributes. Before partitioning, a measure of partial completeness was introduced to quantify information lost due to partitioning and to decide the number of partitions and whether or not to partition a quantitative attribute. The following formula computes the number of required partitions.

$$\text{Number of intervals} = \frac{2n}{m(K-1)} \quad (6)$$

where n is the number of numeric attributes, m is the minimum support and K is the partial completeness level. To identify interesting rules and to prevent the generation of similar rules, the authors used the “greater-than-expected-value” interest measure.

In [14], a novel algorithm, APACS2 was proposed, which implemented *adjusted difference analysis* to find the interesting associations among attributes. This algorithm has the advantage of discovering both positive and negative associations and it avoids user-specified threshold, which is hard to

Table 8 Summary of different numerical association rule mining methods

Methods	Algorithms	Advantages	Disadvantage	Discretization
Optimization method	Evolution based algorithms [7, 10, 20, 39, 45–48, 53, 56, 57, 74]	Strong adaptability and self organization, No need of discretization,	Comparatively higher computational cost	No
	Swarm based algorithms [1, 4, 5, 12, 38, 65, 73]	Efficient, fast convergence rate	Low local search ability	No
	Physics based algorithms [13]	Automatically find the interval of numeric attributes, easy execution	Not very efficient in searching	No
Discretization Method	Clustering [25, 41]	Ability to scale up for high-dimensional cases	Require many user specified threshold, dimension curse	Yes
	Partitioning [14, 22, 60]	Use basic equi-depth and equi-width techniques to the partition which are easy to implement	Challenge to find the best interval, loss of information	Yes
	Fuzzifying [31, 76, 77]	Non-sharp boundaries for interval	Need to choose perfect fuzzy sets	Yes
Distribution Method	Distribution Algorithm [11]	Generate sub rules	No solution for Multiple comparison procedure	Yes

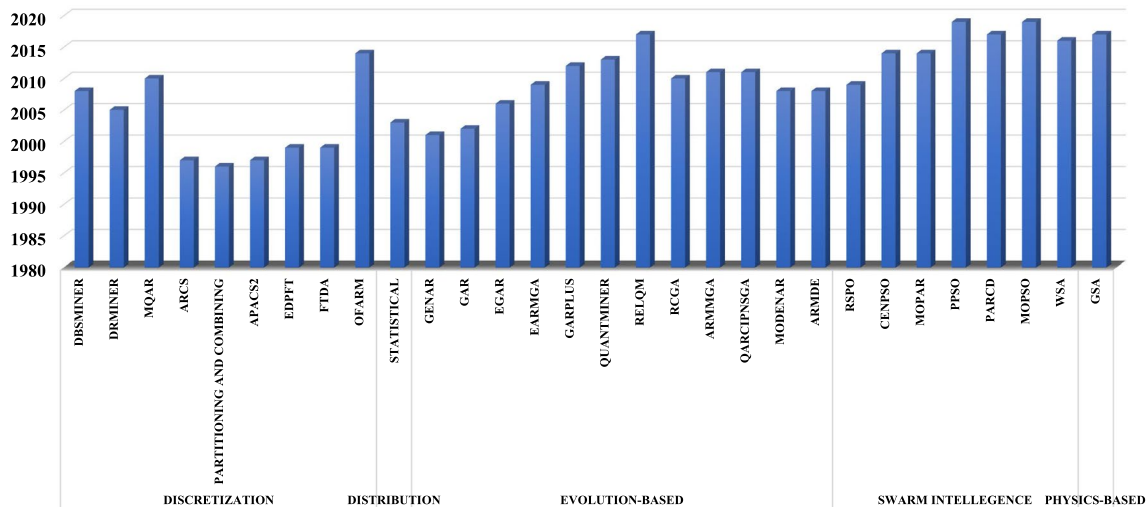


Fig. 2 Year-wise contribution of existing algorithms of NARM

determine. Fukuda et al. [22] presented a novel algorithm to generate optimized intervals in linear time for sorted data. They used *randomized bucketing* as a preprocessing method because it was expensive to sort the quantitative attribute for large databases.

Discussion

In Table 8, we discuss the advantages and disadvantages of the optimization method, discretization, and distribution method. We assessed that every mining method for numerical association rules has some pros and cons. However, being fundamentally different, these approaches have standard support and confidence and mostly have a user-specified threshold. We have investigated that which methods use the discretization technique as a pre-processing step for partitioning or finding the interval of numeric attributes. We observed that all the sub-methods of optimization methods do not use the discretization technique but used in the distribution method. Figure 2 is depicting the year-wise contribution of each method in NARM. It is clear that most of the algorithms of the discretization method were proposed in the 20th century, and few of them were proposed in the 21st century. OFARM is the most recent algorithm that was proposed in 2014. In the swarm intelligence method, parallel PSO and MOPSO are the most recent algorithm among algorithms under all other methods. Algorithms from evolution-based methods came into the scene after 2000. The distribution method was proposed in 2003 and it does not contribute

much to NARM. Recently, a Grand report tool has also been proposed that reports mean values of a chosen numeric target column concerning all possible combinations of influencing factors [51].

Conclusion

Real-world databases contain a high volume of quantitative/numerical and categorical data. Therefore, it is essential to use NARM methods for discovering knowledge from these data sets. In this article, we conducted a detailed study on three NARM methods and their supporting algorithms. We have investigated the use of the discretization technique for partitioning the numerical attributes in various NARM methods. We find that the optimization methods (evolution-based algorithms, swarm-intelligence-based algorithms and physics-based algorithms) do not use discretization techniques; however, they have higher computational costs. The distribution method has not been discussed much in the literature and it does not support the multiple comparison procedure. In the discretization method, dimensionality curse and requirement of many user-specified thresholds is also a disadvantage. Finding the best partition is still very challenging and it has a vast scope in NARM. This article highlighted open research challenges, pros and cons of popular NARM methods and algorithms. We concluded that no single NARM method seems to be perfect for discovering patterns from real-world datasets.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

- Agbehadji IE, Fong S, Millham R. Wolf Search Algorithm for numeric association rule mining. In: IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), IEEE; 2016. pp. 146–51.
- Agrawal R, Imieliński T, Swami A. Mining association rules between sets of items in large databases. *ACM SIGMOD Record*. 1993;22(2):207–16. <https://doi.org/10.1145/170036.170072>.
- Agrawal R, Srikant R. Fast Algorithms for mining association rules in large databases. In: Proceedings of 20th International Conference on Very Large Data Bases, Morgan Kaufmann; 1994 p. 487–99.
- Alatas B, Akin E. Rough particle swarm optimization and its applications in data mining. *Soft Comput*. 2008;12(12):1205–18.
- Alatas B, Akin E. Chaotically encoded particle swarm optimization algorithm and its applications. *Chaos Solit Fract*. 2009;41(2):939–50.
- Alatas B, Akin E. Multi-objective rule mining using a chaotic particle swarm optimization algorithm. *Knowl Based Syst*. 2009;22(6):455–60.
- Alatas B, Akin E, Karci A. Modenar: multi-objective differential evolution algorithm for mining numeric association rules. *Appl Soft Comput*. 2008;8(1):646–56.
- Altay EV, Alatas B. Performance analysis of multi-objective artificial intelligence optimization algorithms in numerical association rule mining. *J Ambient Intell Human Comput*. 2019;2019:1–21.
- Altay EV, Alatas B. Intelligent optimization algorithms for the problem of mining numerical association rules. *Phys A*. 2020;540:123142.
- Álvarez VP, Vázquez JM. An evolutionary algorithm to discover quantitative association rules from huge databases without the need for an a priori discretization. *Expert Syst Appl*. 2012;39(1):585–93.
- Aumann Y, Lindell Y. A statistical theory for quantitative association rules. *J Intell Inf Syst*. 2003;20(3):255–83.
- Beiranvand V, Mobasher-Kashani M, Bakar AA. Multi-objective pso algorithm for mining numerical association rules without a priori discretization. *Expert Syst Appl*. 2014;41(9):4259–73.
- Can U, Alatas B. Automatic mining of quantitative association rules with gravitational search algorithm. *Int J Softw Eng Knowl Eng*. 2017;27(03):343–72.
- Chan KC, Au WH. An effective algorithm for mining interesting quantitative association rules. In: Proceedings of the 1997 ACM symposium on Applied computing; 1997. pp. 88–90.
- Cui Y, Geng Z, Zhu Q, Han Y. Multi-objective optimization methods and application in energy saving. *Energy*. 2017;125:681–704.
- Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Trans Evol Comput*. 2002;6(2):182–97.
- Djenouri Y, Bendjoudi A, Djenouri D, Comuzzi, M. Gpu-based bio-inspired model for solving association rules mining problem. In: 2017 25th euromicro international conference on parallel, distributed and network-based processing (PDP), IEEE; 2017. pp. 262–9.
- Draheim D. Generalized Jeffrey conditionalization: a frequentist semantics of partial conditionalization. Berlin: Springer; 2017.
- Eshelman LJ. The chc adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In: Foundations of genetic algorithms, Elsevier; 1991. vol. 1. pp. 265–83.
- Fister I, Iglesias A, Galvez A, Del Ser J, Osaba E. Differential evolution for association rule mining using categorical and numerical attributes. In: International conference on intelligent data engineering and automated learning, Springer; 2018. pp. 79–88.
- Fonseca CM, Fleming PJ et al. Genetic algorithms for multiobjective optimization: Formulation discussion and generalization. In: Ica, Citeseer; 1993. vol. 93, pp. 416–23.
- Fukuda T, Morimoto Y, Morishita S, Tokuyama T. Mining optimized association rules for numeric attributes. *J Comput Syst Sci*. 1999;58(1):1–12.
- Ghosh A, Nath B. Multi-objective rule mining using genetic algorithms. *Inf Sci*. 2004;163(1–3):123–33.
- Grabmeier J, Rudolph A. Techniques of cluster algorithms in data mining. *Data Min Knowl Disc*. 2002;6(4):303–60.
- Guo Y, Yang J, Huang Y. An effective algorithm for mining quantitative association rules based on high dimension cluster. In: 2008 4th international conference on wireless communications, networking and mobile computing, IEEE; 2008. pp. 1–4.
- Gyenesi A. A fuzzy approach for mining quantitative association rules. *Acta Cybern*. 2001;15(2):305–20.
- Han J, Pei J, Kamber M. Data mining: concepts and techniques. Hoboken: Elsevier; 2011.
- Han J, Pei J, Yin Y, Mao R. Mining frequent patterns without candidate generation: a frequent-pattern tree approach. *Data Min Knowl Disc*. 2004;8(1):53–87.
- Hirasawa K, Okubo M, Katagiri H, Hu J, Murata J. Comparison between genetic network programming (gnp) and genetic programming (gp). In: Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546), IEEE; 2001. vol. 2, pp. 1276–82.
- Holland JH. Adaption in natural and artificial systems. In: An introductory analysis with application to biology, control and artificial intelligence; 1975.
- Hong TP, Kuo CS, Chi SC. Mining association rules from quantitative data. *Intell Data Anal*. 1999;3(5):363–76.
- Kaushik M, Sharma R, Peious SA, Shahin M, Yahia SB, Draheim D. On the potential of numerical association rule mining. In: International conference on future data and security engineering, Springer; 2020. pp. 3–20.
- Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks, IEEE; 1995. vol. 4, pp. 1942–48.
- Khade R, Patel N, Lin J. Supervised dynamic and adaptive discretization for rule mining. In: 2015 In SDM Workshop on Big Data and Stream Analytics; 2015.
- Kianmehr K, Alshalalfa M, Alhaji R. Fuzzy clustering-based discretization for gene expression classification. *Knowl Inf Syst*. 2010;24(3):441–65.
- Kim H, Adeli H. Discrete cost optimization of composite floors using a floating-point genetic algorithm. *Eng Optim*. 2001;33(4):485–501.
- Koza JR, Koza JR. Genetic programming: on the programming of computers by means of natural selection, vol. 1. Berlin: MIT press; 1992.
- Kuo R, Gosumolo M, Zulvia FE. Multi-objective particle swarm optimization algorithm using adaptive archive grid for numerical association rule mining. *Neural Comput Appl*. 2019;31(8):3559–72.
- Kwaśnicka H, Świtalski K. Discovery of association rules from medical data-classical and evolutionary approaches. *Ann Univ Mariae Curie-Sklodowska Sect AI-Inf*. 2006;4(1):204–17.

40. Lent B, Swami A, Widom J. Clustering association rules. In: Proceedings 13th international conference on data engineering, IEEE; 1997. pp. 220–31.
41. Lian W, Cheung DW, Yiu S. An efficient algorithm for finding dense regions for mining quantitative association rules. *Comput Math Appl*. 2005;50(3–4):471–90.
42. Liu H, Abraham A, Li Y, Yang X. Role of chaos in swarm intelligence a preliminary analysis. In: Applications of soft computing, Springer; 2006. pp. 383–92.
43. Liu H, Hussain F, Tan CL, Dash M. Discretization: an enabling technique. *Data Min Knowl Disc*. 2002;6(4):393–423.
44. Lud MC, Widmer G. Relative unsupervised discretization for association rule mining. In: Zighed DA, Komorowski J, Żytkow J, editors. Principles of data mining and knowledge discovery. Berlin, Heidelberg: Springer; 2000. p. 148–58.
45. Martín D, Rosete A, Alcalá-Fdez J, Herrera F. A multi-objective evolutionary algorithm for mining quantitative association rules. In: 2011 11th international conference on intelligent systems design and applications, IEEE; 2011. pp. 1397–402.
46. Martínez-Ballesteros M, Troncoso A, Martínez-Álvarez F, Riquelme JC. Mining quantitative association rules based on evolutionary computation and its application to atmospheric pollution. *Integr Comput-Aided Eng*. 2010;17(3):227–42.
47. Mata J, Alvarez J, Riquelme J. Mining numeric association rules with genetic algorithms. In: Artificial neural nets and genetic algorithms, Springer; 2001. pp. 264–7.
48. Mata J, Alvarez JL, Riquelme JC. Discovering numeric association rules via evolutionary algorithm. In: Pacific-Asia conference on knowledge discovery and data mining, Springer; 2002. pp. 40–51.
49. Mlakar U, Zorman M, Fister I Jr, Fister I. Modified binary cuckoo search for association rule mining. *J Intell Fuzzy Syst*. 2017;32(6):4319–30.
50. Moreland K, Truemper K. Discretization of target attributes for subgroup discovery. In: International workshop on machine learning and data mining in pattern recognition, Springer; 2009. pp. 44–52.
51. Peious SA, Sharma R, Kaushik M, Shah SA, Yahia SB. Grand reports: a tool for generalizing association rule mining to numeric target values. In: International conference on big data analytics and knowledge discovery, Springer; 2020. pp. 28–37.
52. Poli R, Kennedy J, Blackwell T. Particle swarm optimization. *Swarm Intell*. 2007;1(1):33–57.
53. Qodmanan HR, Nasiri M, Minaei-Bidgoli B. Multi objective association rule mining with genetic algorithm without specifying minimum support and minimum confidence. *Expert Syst Appl*. 2011;38(1):288–98.
54. Rashedi E, Nezamabadi-Pour H, Saryazdi S. Gsa: a gravitational search algorithm. *Inf Sci*. 2009;179(13):2232–48.
55. Rashedi E, Rashedi E, Nezamabadi-pour H. A comprehensive survey on gravitational search algorithm. *Swarm Evol Comput*. 2018;41:141–58.
56. Salleb-Aouissi A, Vrain C, Nortet C, Kong X, Rathod V, Cassard D. Quantminer for mining quantitative association rules. *J Mach Learn Res*. 2013;14(1):3153–7.
57. Seki H, Nagao M. An efficient java implementation of a ga-based miner for relational association rules with numerical attributes. In: 2017 IEEE international conference on systems, man, and cybernetics (SMC), IEEE; 2017. pp. 2028–33.
58. Sharma R, Kaushik M, Peious SA, Yahia SB, Draheim D. Expected vs. unexpected: Selecting right measures of interestingness. In: International conference on big data analytics and knowledge discovery, Springer; 2020. pp. 38–47.
59. Shih MY, Jheng JW, Lai LF. A two-step method for clustering mixed categorical and numeric data. *Tamkang J Sci Eng*. 2010;13(1):11–9.
60. Srikant R, Agrawal R. Mining quantitative association rules in large relational tables. In: Proceedings of the 1996 ACM SIGMOD international conference on Management of data; 1996. pp. 1–12.
61. Srinivas N, Deb K. Multiobjective optimization using non-dominated sorting in genetic algorithms. *Evol Comput*. 1994;2(3):221–48.
62. Storn R, Price K. Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces. *J Glob Optim*. 1995;1995:23.
63. Storn R, Price K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim*. 1997;11(4):341–59.
64. Taboada K, Gonzales E, Shimada K, Mabu S, Hirasawa K, Hu J. Association rule mining for continuous attributes using genetic network programming. *IEEE J Trans Electr Electron Eng*. 2008;3(2):199–211.
65. Tahyudin I, Nambo H. The combination of evolutionary algorithm method for numerical association rule mining optimization. In: Proceedings of the tenth international conference on management science and engineering management, Springer; 2017. pp. 13–23.
66. Tan SC. Improving association rule mining using clustering-based discretization of numerical data. In: 2018 international conference on intelligent and innovative computing applications (ICONIC), IEEE; 2018. pp. 1–5.
67. Tang R, Fong S, Yang XS, Deb S. Wolf search algorithm with ephemeral memory. In: Seventh international conference on digital information management (ICDIM 2012), IEEE; 2012. pp. 165–72.
68. Telikani A, Gandomi AH, Shahbahrani A. A survey of evolutionary computation for association rule mining. *Inf Sci*. 2020;2020:5.
69. Triguero I, García S, Herrera F. Differential evolution for optimizing the positioning of prototypes in nearest neighbor classification. *Pattern Recogn*. 2011;44(4):901–16.
70. Webb GI. OPUS: An efficient admissible algorithm for unordered search. *J Artif Intell Res*. 1995;3:431–65. <https://doi.org/10.1613/jair.227>
71. Webb GI. Discovering associations with numeric variables. In: Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2001; pp. 383–8.
72. Yamany W, Emary E, Hassanien AE. Wolf search algorithm for attribute reduction in classification. In: IEEE Symposium on Computational Intelligence and Data Mining (CIDM), IEEE. 2014; pp. 351–8. <https://doi.org/10.1109/CIDM.2014.7008689>
73. Yan D, Zhao X, Lin R, Bai D. Ppqar Parallel pso for quantitative association rule mining. *Peer-to-Peer Netw Appl*. 2019;12(5):1433–44.
74. Yan X, Zhang C, Zhang S. Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support. *Expert Syst Appl*. 2009;36(2):3066–76.
75. Yang J, Feng Z. An effective algorithm for mining quantitative associations based on subspace clustering. In: International Conference on Networking and Digital Society IEEE; 2010;1:175–8.
76. Zhang W. Mining fuzzy quantitative association rules. In: Proceedings of 11th International Conference on Tools with Artificial Intelligence, IEEE; 1999. pp. 99–102.
77. H. Zheng, J. He, G. Huang and Y. Zhang. Optimized fuzzy association rule mining for quantitative data. In: IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE; 2014. pp. 396–403. <https://doi.org/10.1109/FUZZ-IEEE.2014.6891735>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.