Differential evolution for association rule mining using categorical and numerical attributes

Iztok Fister Jr.[1⋆], Andres Iglesias[2,3], Akemi Galvez[2,3], Javier Del Ser[4,5,6], Eneko Osaba[4], and Iztok Fister[1,3]

[1] University of Maribor, Faculty of Electrical Engineering and Computer Science, Koroška cesta 46, Slovenia
[2] Toho University, 2-2-1 Miyama, 274-8510, Funabashi, Japan
[3] University of Cantabria, Avenida de los Castros, s/n, 39005 Santander, Spain
[4] TECNALIA, Derio, Spain
[5] University of the Basque Country (UPV/EHU), Bilbao, Spain
[6] Basque Center for Applied Mathematics (BCAM), Bilbao, Spain

Abstract. Association rule mining is a method for identification of dependence rules between features in a transaction database. In the past years, researchers applied the method using features consisting of categorical attributes. Rarely, numerical attributes were used in these studies. In this paper, we present a novel approach for mining association based on differential evolution, where features consist of numerical as well as categorical attributes. Thus, the problem is presented as a single objective optimization problem, where support and confidence of association rules are combined into a fitness function in order to determine the quality of the mined association rules. Initial experiments on sport data show that the proposed solution is promising for future development. Further challenges and problems are also exposed in this paper.

Keywords: Association rule mining, Classification, Differential evolution, Evolutionary computation

1 Introduction

Data mining methods are intended to infer new insights (knowledge) from a bunch of structured or unstructured data. With the rise of big data on almost all areas of human endeavor, data mining methods have received a high priority within business and industry. In the past, many methods for various data mining tasks were developed, which are primarily devoted for the classification, clustering, regression as well as association rule mining (ARM). ARM is basically the process of identifying the dependence rules between features inside transaction databases. Therefore, ARM is appropriate for market basket analysis, analysis of human habits, driver strategies and so on.

Researchers introduced association rule mining in the 90s with seminal work of Agrawal [3]. From that time, ARM was deeply studied in theory and practice.

⋆ Corresponding Author: iztok.fister1@um.si

After more than 20 years of research, these methods have became matured. On the other hand, rising of the big data has demanded development of the new scalable methods that can easily be parallelized. There exist many ARM algorithms, as for example Apriori [3], Eclat [19], FP-growth [12].

Interestingly, researchers also developed some ARM methods that are based on stochastic population-based nature-inspired algorithms [4, 6, 7, 16], such as Particle Swarm Optimization (PSO) [14], Ant Colony Optimization (ACO) [8], Differential Evolution (DE) [17]. Main features of these algorithms is that they can easily be parallelized, while they basically also ensure scalability.

However, ARM was initially applied for mining categorical (i.e., non-numerical) features, where all features must be discretized before usage. Normally, this task is trivial due to a simple mapping of the attribute to intervals of real values representing the corresponding attributes. However, similar methods that are able to handle also numeric attributes were proposed in [11, 5, 15, 2]. Introducing the numerical attributes demands dealing with explicit intervals of real values for each numerical attribute of the feature.

Therefore, the purpose of the paper is to extend our previous work on ARM for mining categorical features based on bat algorithm, i.e. BatMiner [10] with new feature that allows mining combination of numerical data as well as categorical data. Additionally, the bat algorithm [18] is here replaced by differential evolution. There are two issues for this decision as follows: (1) To show how the evolutionary algorithms behaves by solving this problem, and (2) To find out how the operator of crossover affects the results of the mining. As already known, the bat algorithm works only using the mutation operator [9]. Algorithm is evaluated on sport data that consists of 14 features, where attributes of three features are numerical, while the attributes of the 11 other features represent categorical values.

In a nutshell, main contributions of this paper are as follows:

- A new differential evolution algorithm for mining association rules is proposed.
- The algorithm is capable of dealing with numerical and categorical features.
- The algorithm is tested on dataset that encompass habits of athlete in training.

The structure of this paper is as follows: In Section 2, the background information is discussed that is necessary for understanding the subjects in the remainder of the paper. Section 3 describes the proposed algorithm for ARM using mixed numerical and categorical feature attributes. In Section 4, experiments and results are illustrated, while the paper is concluded with summarizing the performed work and outlining the possible directions for the future.

2 Background information

2.1 Association rule mining

This section briefly presents formal definition of ARM. Let us suppose, a set of objects $O = \{o_1, \ldots, o_n\}$ and transaction set D are given, where each transaction

T is a subset of objects, in other words $T \subseteq O$. Then, an association rule can be defined as 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 association rule [3]:

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

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

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

2.2 Differential evolution basics

Differential Evolution is an Evolutionary Algorithm appropriate for continuous as well as combinatorial optimization. This algorithm was introduced by Storn and Price in 1995 [17]. It is a population-based and consists of Np real-coded vectors representing the candidate solutions, as follows:

$$\mathbf{x}_i^{(t)} = (x_{i,1}^{(t)}, \ldots, x_{i,n}^{(t)}), \quad \text{for } i = 1, \ldots, Np, \tag{4}$$

where each element of the solution is in the interval $x_{i,1}^{(t)} \in [x_i^{(L)}, x_i^{(U)}]$, and $x_i^{(L)}$ and $x_i^{(U)}$ denotes the lower and upper bounds of the i-th variable, respectively.

The variation operator in DE supports a differential mutation and a differential crossover. In particular, the differential mutation selects two solutions randomly and adds a scaled difference between these to the third solution. This mutation is expressed as follows:

$$\mathbf{u}_i^{(t)} = \mathbf{x}_{r1}^{(t)} + F \cdot (\mathbf{x}_{r2}^{(t)} - \mathbf{x}_{r3}^{(t)}), \quad \text{for } i = 1, \ldots, Np, \tag{5}$$

where F denotes the scaling factor as a positive real number that scales the rate of modification while $r1,\ r2,\ r3$ are randomly selected values in the interval $1 \ldots Np$. Note that, typically, the interval $F \in [0.1, 1.0]$ is used in the DE community.

As a differential crossover, uniform crossover is employed by the DE, where the trial vector is built from parameter values copied from two different solutions. Mathematically, this crossover is expressed as follows:

$$w_{i,j}^{(t+1)} = \begin{cases} u_{i,j}^{(t)} & \text{rand}_j(0,1) \leq CR \vee j = j_{rand}, \\ x_{i,j}^{(t)} & \text{otherwise}, \end{cases} \tag{6}$$

where $CR \in [0.0, 1.0]$ controls the fraction of parameters that are copied to the trial solution. Note, the relation $j = j_{rand}$ ensures that the trial vector is different from the original solution $\mathbf{x}_i^{(t)}$.

A differential selection is, in fact, a generalized one-to-one selection that is expressed mathematically as follows:

$$\mathbf{x}_i^{(t+1)} = \begin{cases} \mathbf{w}_i^{(t)} & \text{if } f(\mathbf{w}_i^{(t)}) \leq f(\mathbf{x}_i^{(t)}), \\ \mathbf{x}_i^{(t)} & \text{otherwise}\,. \end{cases} \tag{7}$$

In a technical sense, crossover and mutation can be performed in several ways in Differential Evolution. Therefore, a specific notation is used to describe the varieties of these methods (also strategies) generally. For example, 'DE/rand/1/bin' denotes that the base vector is selected randomly, 1 vector difference is added to it, and the number of modified parameters in the mutant vector follows binomial distribution.

3 DE for ARM using numerical and categorical attributes

In this section, a new DE for ARM using mixed (i.e., numerical and categorical) attributes (ARM-DE) is proposed. Basically, development of this algorithm consists of the following stages:

- domain analysis,
- representation of solution,
- definition of a fitness function.

The aim of the first stage is the identification of features. In the second stage, the representation of solution is discussed, while a definition of the fitness function is illustrated in the last stage. In the remainder of the paper, the mentioned stages are presented in detail.

3.1 Domain analysis

In this stage, we identify the features along with their attributes and corresponding domain values. Actually, there can be numerical as well as categorical attributes. If attributes are numerical values, minimum lower and maximum upper bounds are prescribed determining intervals of domain values, from which the minimum and maximum values for each particular numerical attribute can be drawn. For categorical attributes, a set of feasible attributes needs to be enumerated, because of their discrete nature.

The minimum lower and maximum upper bounds, from which the numerical attributes can be drawn, are presented in Table 1. There are also eleven features with categorical attributes as illustrated in Table 2. The numerical attributes are presented in the tables as a particular feature with corresponding interval of domain values (i.e., DISTANCE $\in [50, 100]$), while the categorical attributes as the feature followed by an attribute preceded by point sign '.' (i.e., CALORIES.SMALL).

Table 1. Numerical attributes and their domain values.

Feature	Attribute domain	
	Minimum lower bound	Maximum upper bound
DISTANCE	20	200
DURATION	20	330
HEART_RATE	80	185

Table 2. Categorical attributes and their domain values.

Feature	Attribute domain
CALORIES	{NULL, SMALL, MEDIUM, HIGH}
WEATHER	{NULL, SUNNY, CLOUDY, RAINY, SNOWY}
TYPE	{NULL, EASY, INTERVALS, POWER, ENDURANCE}
NUTRITION	{NULL, POOR, MODERATE, GOOD}
FOOD	{NULL, PROTEINS, CARBOHYDRATES, FAT, FRUITS}
BEVERAGES	{NULL, WATER, JUICE, ISO, COKE}
REST	{NULL, AFTER_TRAINING, NO}
NIGHT_REST	{NULL, BAD, MEDIUM, GOOD}
INJURIES	{NULL, NO, LOW, MEDIUM, HIGH}
CRAMPS	{NULL, NO, LOW, HIGH}
HEALTH_PROBLEMS	{NULL, NO, LITTLE, YES}

3.2 Representation of solutions

Each individual is represented as a real-valued vector, where every numerical attribute is represented by a corresponding minimum and maximum boundaries determining domain values, from which the attribute can be drawn. In contrary, every categorical attribute is represented by the real-value drawn from interval $[0, 1]$. Additionally, the last element in the vector denotes the cut point. It means that this number says, which part of the vector belong to the antecedent and which to the consequence of the mined association rule.

In a nutshell, each solution is mathematically represented as the real-valued vector:

$$\mathbf{x}_i^{(t)} = \{\underbrace{(x_{i,1}^{(t)}, x_{i,2}^{(t)})}_{Attr_1^{(num)}}, \ldots, \underbrace{(x_{i,2n-1}^{(t)}, x_{i,2n}^{(t)})}_{Attr_n^{(num)}}, \underbrace{x_{i,2n+1}^{(t)}}_{Attr_{n+1}^{(cat)}}, \ldots, \underbrace{x_{i,d}^{(t)}}_{Attr_{d-n}^{(cat)}}, \underbrace{x_{i,d+1}^{(t)}}_{\text{Cut point}}\}, \quad (8)$$

where $x_{i,j}^{(t)}$ for $i = 1, \ldots, d$ codes attributes of features in association rule, $x_{i,d+1}^{(t)}$ denotes cut point and t is counter of iterations. Thus, each numerical attribute is represented as a pair $Attr_j^{(num)} = (x_{i,2j-1}^{(t)}, x_{i,2j}^{(t)})$ for $j = 1, \ldots, n$, where $x_{i,2j-1}^{(t)} \in [Lb_{2j-1}]$ and $x_{i,2j}^{(t)} \in [Lb_{2j}]$, and the attribute is calculated according to the following equation:

$$Attr_j^{(num)} = \begin{cases} (NULL, NULL), \text{ if } |x_{i,2j-1}^{(t)} - x_{i,2j}^{(t)}| < \lfloor \frac{Ub_{2j-1} - Lb_{2j-1}}{50} \rfloor, \\ (x_{i,2j-1}^{(t)}, x_{i,2j}^{(t)}), \quad \text{if } x_{i,2j-1}^{(t)} > x_{i,2j}^{(t)}, \\ (x_{i,2j}^{(t)}, x_{i,2j-1}^{(t)}), \quad \text{otherwise.} \end{cases} \quad (9)$$

In Eq. 9, the Lb_{2j-1} and Ub_{2j-1} denote the lower and upper bound of the specific attribute value, respectively. On the other hand, the range of feasible values for categorical attributes in interval $x_{i,j}^{(t)} \in [0,1]$ is divided into $m_j + 1$ equidistant intervals, where each interval $[k, k+1)$ for $k = 0, \ldots, m_j$ corresponds to one of the potential attributes $Attr_j^{(cat)} \in \{Attr_0, \ldots, Attr_{m_j}\}$, and parameter m_j denotes the number of attributes belonging to the j-th feature. Actually, the categorical attribute is calculated according to the following equation:

$$Attr_j^{(cat)} = \lfloor \frac{x_{i,j}^{(t)}}{m_j + 1} \rfloor, \text{for } j = n, \ldots, d - n. \tag{10}$$

Attribute $Attr_0^{(cat)} = \text{NULL}$ has a special meaning, because it determines that the corresponding feature is not presented in the association rule.

Finally, the cut point is calculated according to the following equation:

$$cp_i^{(t)} = \lfloor x_{i,d+1}^{(t)} \cdot (d - n - 2) \rfloor + 1, \text{for } i = 0, \ldots, Np, \tag{11}$$

3.3 Definition of the fitness function

Fitness function is defined as follows [10, 13]:

$$f(\mathbf{x}_i^{(t)}) = \begin{cases} \alpha * conf(\mathbf{x}_i^{(t)}) + \gamma * supp(\mathbf{x}_i^{(t)})/\alpha + \gamma, & \text{if } feasible(\mathbf{x}_i^{(t)}) = true, \\ -1, & \text{drugae}, \end{cases} \tag{12}$$

where $conf()$ is confidence, $supp()$ support, α and γ are weights, function $feasible(\mathbf{x}_i)$, denotes if solution is feasible. The task of optimization is to find maximum value of fitness function.

4 Experiments and results

The aim of experimental work was to test the performance of proposed ARM-DE in practice. In line with this, it is expected that the results of this algorithm make sense from the real sports trainer point of view. As mentioned before, ARM-DE is able to handle both categorical and numerical data stored in transaction database consisting of 14 features, where the first three features have numerical attributes (i.e., distance, duration and average heart rate), while the other eleven feature's attributes are represented by categorical data.

Let us mention that we taken the full set of attributes that can be obtained from the sport activity datasets. Obviously, each sport activity presents one transaction in transaction database. Unfortunately, the real data is hard to obtain. In the case, when the real activities can be gained from the athlete, the number of these activities is limited due to physical limitation of an athlete. The professional cyclist, for example, finishes typically one sports training per day. When we assume that one day is reserved for resting, this athlete can perform at

most 300 sports activities per year. This number is rapidly decreased, if the amateur cyclist is taken into consideration that cannot be affordable to train each day. Therefore, the generator of sports training activities has been proposed by Fister et al. [1], with which these limitations can be circumvent. The generated sports activities were used also in our study.

Because of lack of real data, the transaction database consisting of 500 transactions were generated randomly, in this study. The following parameters were used in DE algorithm: $D = 19$, $NP = 100$, $FES = 70,000$, $F = 0.5$, $CR = 0.9$, where FES represents the number of fitness function evaluations. Thus, all feasible solutions obtained after 10 independent runs of the ARM-DE algorithm were accumulated, while the best results are presented in Table 3, and their corresponding quality measures in Table 4. Let us mention, that all numerical attributes in our experiments must be present in each association rule, otherwise this is considered as infeasible. Illustrated association rules referring to athlete's

Table 3. Examples of the best solutions found by the ARM-DE algorithm.

Rule	Antecedent	Consequent
1	DISTANCE $\in [25.80, 111.96] \wedge$ DURATION $\in [31.08, 172.86] \wedge$ HEART_RATE $\in [136.68, 180.52) \wedge$ WEATHER.SNOWY	REST.NO
2	DISTANCE $\in [133.32, 220] \wedge$ DURATION $\in [281.10, 315.19]$ HEART_RATE $\in [104.62, 149.49) \wedge$ TYPE.ENDURANCE	HEALTH_PROBLEMS.NO
3	DISTANCE $\in [160.14, 220] \wedge$ DURATION $\in [181.22, 330.0]$ HEART_RATE $\in [111.47, 139.56) \wedge$ TYPE.INTERVALS	HEALTH_PROBLEMS.NO
4	DISTANCE $\in [121.37, 199.40] \wedge$ DURATION $\in [187.66, 330]$ HEART_RATE $\in [133.02, 168.80) \wedge$ BEVERAGES.JUICE $\wedge$ REST.AFTER_TRAINING	HEALTH_PROBLEMS.NO

Table 4. Quality measures obtained by the best solutions found.

Rule	Fitness	Support	Confidence
1	0.519	0.038	1.0
2	0.516	0.032	1.0
3	0.512	0.024	1.0
4	0.511	0.022	1.0

health condition are very similar between each other and say that cyclist overcoming moderate-distance courses in ultra-long duration and moderate intensity, in either endurance or interval type of training session, or drinking juice during and resting after the training probably should not have any health problems. Obviously, the rule is valid in real-world.

Similar as observed in our previous work [10], the best solutions usually have only one consequence. However, there are also more complex association rules in Table 5 with their corresponding quality measures in Table 6, where more

attributes can be observed in association rules. The last association rule in the

Table 5. Examples of more complex solutions that were found by ARM-DE algorithm.

Rule	Antecedent	Consequent
1	DISTANCE $\in [20.0, 71.39] \wedge$ DURATION $\in [178.55, 139.89] \wedge$ HEART_RATE $\in [139.89, 180.18) \wedge$ CALORIES.HIGH $\wedge$ TYPE.ENDURANCE	CRAMPS.NO $\wedge$ HEALTH_PROBLEMS.NO
2	DISTANCE $\in [85.03, 134.63] \wedge$ DURATION $\in [81.81, 228.50] \wedge$ HEART_RATE $\in [93.96, 146.05) \wedge$ NUTRITION.GOOD $\wedge$ FOOD.PROTEINS $\wedge$ BEVERAGES.JUICE $\wedge$ REST.AFTER_TRAINING	CRAMPS.NO $\wedge$ HEALTH_PROBLEMS.NO
3	DISTANCE $\in [20.00, 86.60] \wedge$ DURATION $\in [155.31, 161.87] \wedge$ HEART_RATE $\in [161.87, 183.43) \wedge$ CALORIES.HIGH $\wedge$ TYPE.POWER	INJURIES.NO $\wedge$ CRAMPS.NO

Table 6. Quality measures obtained by the more complex solutions found.

Rule	Fitness	Support	Confidence
1	0.505	0.010	1.0
2	0.503	0.006	1.0
3	0.502	0.004	1.0

table is the most interesting and asserts that cyclist overcoming the medium-distance course in ultra-long duration with moderate intensity and high calorie consumption by performing the power training session should not have the injuries as well as cramps. This rule also hold in practice.

5 Conclusion and future challenges

Association rule mining with numerical attributes is a very challenging problem. In this paper, we try to solve the problem using differential evolution. Our proposed solution is capable to mine association rule, where features can consist of either numeric or categorical attributes. This assertion is evident by obtaining the highest confidence values of mined rules. Practical experiments on synthetically generated data showed that this approach is interesting, but there are still many problems that should be elaborated in future, i.e. how to shrink lower and upper borders on numerical attributes, how to better evaluate these borders, testing the algorithm on bigger transaction databases with more features as well as applying this approach to other population-based nature-inspired algorithms.

Acknowledgment

I. Fister Jr. and I. Fister acknowledge the financial support from the Slovenian Research Agency (Research Core Fundings No. P2-0041 and P2-0057). A. Iglesias and A. Galvez acknowledge the financial support from the projects #TIN2017-89275-R (AEI/FEDER, UE), and #JU12 (SODERCAN/FEDER UE). E. Osaba and J. Del Ser would like to thank the Basque Government for its funding support through the EMAITEK program.

References

1. SportyDataGen. http://www.sport-slo.net/. Accessed: 2018-05-30.
2. I. E. Agbehadji, S. Fong, and R. Millham. Wolf search algorithm for numeric association rule mining. In *Cloud Computing and Big Data Analysis (ICCCBDA), 2016 IEEE International Conference on*, pages 146–151. IEEE, 2016.
3. R. Agrawal, R. Srikant, et al. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499, 1994.
4. B. Alataş and E. Akin. An efficient genetic algorithm for automated mining of both positive and negative quantitative association rules. *Soft Computing*, 10(3):230–237, 2006.
5. B. Alatas and E. Akin. Rough particle swarm optimization and its applications in data mining. *Soft Computing*, 12(12):1205–1218, 2008.
6. B. Alatas and E. Akin. Multi-objective rule mining using a chaotic particle swarm optimization algorithm. *Knowledge-Based Systems*, 22(6):455–460, 2009.
7. F. Chiclana, R. Kumar, M. Mittal, M. Khari, J. M. Chatterjee, S. W. Baik, et al. Arm-amo: An efficient association rule mining algorithm based on animal migration optimization. *Knowledge-Based Systems*, 2018.
8. M. Dorigo and C. Blum. Ant colony optimization theory: A survey. *Theoretical Computer Science*, 344(2):243 – 278, 2005.
9. I. Fister, J. Brest, and U. Mlakar. Towards the universal framework of stochastic nature-inspired population-based algorithms. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8, Dec 2016.
10. I. Fister Jr. and I. Fister. Algoritem batminer za rudarjenje asociativnih pravil. *Presek*, 44(5):26–29, 2017.
11. T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Mining optimized association rules for numeric attributes. In *Proceedings of the fifteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 182–191. ACM, 1996.
12. J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *ACM sigmod record*, volume 29, pages 1–12. ACM, 2000.
13. K. E. Heraguemi, N. Kamel, and H. Drias. Association rule mining based on bat algorithm. In *Bio-Inspired Computing-Theories and Applications*, pages 182–186. Springer, 2014.
14. J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948. IEEE, 1995.
15. B. Minaei-Bidgoli, R. Barmaki, and M. Nasiri. Mining numerical association rules via multi-objective genetic algorithms. *Information Sciences*, 233:15–24, 2013.

16. U. Mlakar, M. Zorman, I. Fister Jr, and I. Fister. Modified binary cuckoo search for association rule mining. *Journal of Intelligent & Fuzzy Systems*, 32(6):4319–4330, 2017.
17. R. Storn and K. Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.
18. X.-S. Yang. *Nature-inspired metaheuristic algorithms*. Luniver press, 2010.
19. M. J. Zaki. Scalable algorithms for association mining. *IEEE transactions on knowledge and data engineering*, 12(3):372–390, 2000.