Association Rule Mining via Evolutionary Multi-objective Optimization

Pradeep Ganghishetti[1,2] and Ravi Vadlamani[1,*]

[1] Centre of Excellence in CRM & Analytics,
Institute for Development and Research in Banking Technology,
Hyderabad-500 057, A.P, India
[2] SCIS, University of Hyderabad, Hyderabad-500046, A.P, India
pradeepghyd@gmail.com, rav_padma@yahoo.com

Abstract. In this paper, we formulated association rule mining as a combinatorial, multi-objective global optimization problem by considering measures such as support, confidence, coverage, comprehensibility, leverage, interestingness, lift and conviction. Here, we developed three evolutionary miners viz., Multi-objective Binary Particle Swarm Optimization based association rule miner (MO-BPSO), a hybridized Multi-objective Binary Firefly Optimization and Threshold Accepting based association rule miner (MO-BFFOTA), hybridized Multi-objective Binary Particle Swarm Optimization and Threshold Accepting based association rule miner (MO-BPSOTA) and applied them on various datasets and conclude that MO-BPSO-TA outperforms all others .

Keywords: Evolutionary Multi-objective Optimization, Association rules, Crisp Product Operator, Combinatorial Global Optimization, Quality Measures.

1 Introduction

Association rule mining is a very important data mining task used to extract important correlations among the products from transactional databases [1]. An association Rule is of the form A→B, where A and B represent item sets (I) or products, and an item set includes all possible items$\{i_1, i_2, \ldots, i_m\}$ in a transactional database. The a priori algorithm [2] works in two phases- generation of frequent item-sets and rule generation. Later, Han et al. [3] proposed FP-Growth algorithm that generates a F-List and a tree followed by mining of rules. Association rules have been extracted by leveraging evolutionary computational techniques. Saggar et al. [4] optimizes association rules extracted by a priori via Genetic Algorithm (GA). Anandhavalli et al. [5] also uses GA based approach for extracting association rules with negative attributes. Multi-objective GA based approaches have also been suggested [6-8]. Kaya et al.[9] extracted partial optimized fuzzy association rules. Then, particle swarm optimization (PSO) was used for association rule mining [10,11]. Nandhini et al. [12] developed domain ontology and PSO based association rule mining algorithm. Alatas et al. [13]

[*] Corresponding author.

M.N. Murty et al. (Eds.): MIWAI 2014, LNAI 8875, pp. 35–46, 2014.

devised pareto-based multi-objective differential evolution (DE) for extracting association rules. Kuo et al. [14] used PSO to objectively and quickly determine user-defined parameters. Menéndez et al [15-17] proposed Multi-objective Genetic Algorithm (MOGGC) for spectral Clustering problem, feature selection followed by clustering and image segmentation. Most recently, Sarath and Ravi [18] develooped binary PSO to extract association rules from datasets without having to specify the minimum support and confidence upfront. Naveen et al. [19] devised firefly optimization based rule extractor for classification Maheshkumaret al [20] proposed PSO-TA hybridized algorithm for solving unconstrained continuous optimization.

2 Various Optimization Based Techniques

2.1 Firefly Optimization Algorithm (FFO)

The Firefly algorithm [21] is a population based global optimization technique inspired by the natural behavior of fireflies. Each firefly moves to more brighter/attractive firefly wherein firefly's brightness is characterized by its objective function. The attractiveness of a firefly is directly proportional to its brightness and decreases as the distance from the other firefly increases. Here each firefly represents a solution in the optimization parlance. The attractiveness (β) is a monotonically decreasing function of the distance r between any two fireflies.

$$\beta = \beta_0 e^{-\gamma r^2} \tag{1}$$

The traversal of a firefly towards other brighter fireflies is given by:

$$x_i(t+1)=x_i(t)+\beta_0 e^{-\gamma r^2}\left(x_i-x_j\right)+\alpha(\mathrm{rand}()-0.5) \tag{2}$$

The algorithm of the firefly optimization is depicted in the box as follows:

Algorithm:
- Define Objective function f(x), $x=(x_1,\dots x_d)^T$, β_0, γ
- Generate initial population of fireflies X_i (i=1,2,…,n)
- Light intensity L_i at x_i is determined by $f(x_i)$
- While(t<MaxGeneration)
- for i=1:n all n fireflies
-     for j=1:n all n fireflies
-         if(L_j>L_i), Move firefly i towards j in d-dimensions as in eqn 1 and 2 end if
-     Evaluate new solutions & update light intensity
-     end for j
- end for i
- Rank the fireflies and find the current best
- end while

where t denotes iteration number, rand() indicates a random number between 0 and 1, β_0 denotes the attractiveness constant and γ is the light absorption coefficient. After every iteration, randomness parameter (α) is reduced by a constant Delta (Δ). The distance between two fireflies i and j at positions x_i and x_j can be defined as follows:

$$r_{ij} = \left\|x_i - x_j\right\| = \sqrt{\sum_{k=1}^{d}(x_{ik} - x_{jk})^2} \tag{3}$$

2.2 Threshold Acceptance Algorithm (TA)

Threshold Accepting algorithm, proposed by Deuck and Sheuer [22] is a point that is not much worse than the current one. A candidate solution Cand_Soln[i] in the neighborhood following rule scheme presented in Table 1 is generated as follows:

```
Algorithm:
•Consider weakest particle/firefly of MO-BPSO/MO-BFFO as the initial solution and let its fitness be fi.
•for each of the global iterations
•       While(innerIterationNo<MaxInnerIterations or Δ1 > thresh)
•         Generate a candidate solution in the neighborhood and let its fitness be fj.
•         Compute deterioration Δ1=fi-fj
•         If Δ1 < thresh, set fi = fj                  //To accept/reject the solution
•       If thresh <thrtol, set Δ2 = (new - old) / old  //for execution of ample no. of global iters
•         Report current solution as the optimal one if abs(Δ2) <acc and exit if end of global iters
•       Else
•         old=new
```

```
for(j=0;j<2×n;j++)                      //generation of a candidate solution
        If(rand()<bias)       Cand_Soln[i][j]=1
        else        Cand_Soln[i][j]=0
```

Table 1. Rule representation

I_1		I_2		I_3		…	I_N	
V_{11}	V_{12}	V_{21}	V_{22}	V_{31}	V_{32}	…	V_{N1}	V_{N2}

Likewise, we generate rules of varied lengths (like 2,3,4,..). If there are 25 features in the dataset, the bias value of 0.1 is found to be suitable after several computations. The bias to be fixed is inversely proportional to the number of features in the dataset ($bias_1 \times featuresCount_1 = bias_2 \times featuresCount_2$).

2.3 Binary Firefly Optimization (MO-BFFO)

We developed a combinatorial version of MO-BFFO to solve our optimization problem. Here, each firefly's positional value in each dimension can be either 0 or 1 only. This value of x_i is determined probabilistically depending on the changing rate of real value of x_i. For binary version of firefly optimization, the firefly's positions are updated as follows:

$$x_{i+1} = x_i + \beta_0 e^{-\gamma r^2}(x_i - x_j) + \alpha(rand - 0.5)$$

If (rand () < S (x_{i+1}))
Then $x_{i+1}=1$
Else $x_{i+1}=0$

Where S is the sigmoid function, rand () is the uniform random number between (0, 1), β_0 is the attractiveness constant, γ is the absorption coefficient and r is the Euclidean distance between the two fireflies i and j. The sigmoid function used in our algorithms is as follows: $S(x) = \frac{1}{1+e^{-x}}$. Binary PSO was developed in [18] as a binary version of PSO [23]. For more details the reader is referred to [18].

3 Multi-objective Association Rule Miners

3.1 Preprocessing

The first step involves transformation of data into binary encoded format where each record is stored in terms of 0's and 1's [24] as in fig 1. This step is necessary for faster database scanning and faster calculation of various measures. Let there be transactions T_1-T_5 with five different items I_1-I_5. For instance, in transaction T_5, the values of cells I_1 and I_5 are both "1's" whereas cells I_2, I_3, I_4 are "0's" indicating items I_1 and I_5 are only purchased. The second step of preprocessing is feature selection used when the number of features is above 50. Here, we discard all the features whose item support is less than α, where α is very small user defined value. This helps in obviating the unnecessary computations and removing extremely rare rules.

T_1	I_2	I_3		
T_2	I_1	I_2	I_5	
T_3	I_1	I_3	I_4	I_5
T_4	I_1	I_2		
T_5	I_2	I_5		

	I_1	**I_2**	**I_3**	**I_4**	**I_5**
B_1	0	1	1	0	0
B_2	1	1	0	0	1
B_3	1	0	1	1	1
B_4	1	1	0	0	0
B_5	0	1	0	0	1

Fig. 1. Binary Transformation of the original dataset

3.2 Rule Representation

In this paper, we followed Michigan approach [25], where each chromosome represents a separate rule. Let there be N number of items in the dataset. Each item is represented by two bits and each bit can take the values either 1/0. The value of 1 in the first bit indicates the item is present otherwise absent in the rule. The second bit signifies whether the item is included in the antecedent (i.e., 1) or consequent (i.e., 0).

11-item present in antecedent   10-item present in consequent   00/01-item absent in the rule

3.3 Quality Measures

In most cases, the quality of association rules cannot be judged by considering only support and confidence but there are other measures as follows. Let a rule be represented as A→B where A is the antecedent and B is called the consequent.

1. Support
Support is defined as the percentage or fraction of transactions in the database that contain both antecedent as well as the consequent parts.

2. Confidence:
Confidence indicates how **reliable or relevant** a given rule is. Confidence is defined as the probability of occurring the rule's consequent under the condition that the transactions also contain the antecedent.

3. Interestingness:
A rule is said to be interesting when the individual support count values are greater than the collective support (A→B) values.

$$Interestingness = \frac{Support(AUB)}{Support(A)} \times \frac{Support(AUB)}{Support(B)} \times \left(1 - \frac{Support(AUB)}{Support(D)}\right)$$

4. Comprehensibility:
In association rule mining, if the number of conditions involved in the antecedent part is less than the one in the consequent part, the rule is more comprehensible.

Comprehensibility=log(1+|c|/log(s+|A U C|))

5. Lift:

$$Lift(A \to B) = \frac{Confidence(A \to B)}{Support(B)} = \frac{Support(A) \times Support(B)}{Support(B)}$$

The lift value is a measure of **importance** of a rule (originally called interest)

6. Leverage:
The rationale in a sales setting is to find how many more units (items X and Y together) are sold than expected from the independent sells and shows the impact of ARM.

$$Leverage(A \to B) = Support(A \to B) - Support(A) \times Support(B)$$

7. Conviction:

$$Conviction(A \to B) = \frac{\left(1 - Support(B)\right)}{\left(1 - Confidence(A \to B)\right)} = \frac{P(A)P(!\,B)}{P(Aand\,!\,B)}$$

Conviction compares the probability that Antecedent appears without Consequents if they were dependent with the actual frequency of the appearance of Antecedents without Consequents.

8. Coverage

$$Coverage = Support(A)$$

Coverage measures how often a rule X -> Y is applicable in a database. Objective function

The choice of fitness/objective function is helpful in assessing the importance of each individual firefly in the population set. Here, we assign equal weightage to all the above defined measures and hence use the product of all of them to compute the objective function as follows:

$$\text{Fitness} = \prod_{i=1}^{8} Measure\_Value(i) \tag{4}$$

3.4 Steps Involved in MO-BPSO

In MO-BPSO, we proposed the fitness our proposed model, we need to run the algorithm M times in order to get all the top M rules in the database. In every run, we store the positions and movements of all the fireflies in the population for each of the iterations. After all the runs, the records are ranked and duplicate records are removed. We pick the top M rules as per the objective function. The main idea behind running the algorithm M times is as follows. Whenever evolutionary algorithms are employed to classification or association rule mining problem, the user gets different

Table 2. Parameters chosen for MO-BPSO Rule Miner

Dataset	N	Inertia	C1	C2	Max Iterations
Book	30	0.8	2	2	50
Food	30	0.8	2	2	50
Grocery	30	0.8	2	2	100
Bank	50	0.8	2	2	50
Click stream	30	100.8	2	2	50
Bakery	30	0.8	2	2	50

rules in every run. Therefore, running the MO-BPSO many times and collating the rules obtained indeed becomes an ensembling scheme for extracting rules. The parameters used in MO-BPSO are presented in Table 2.

3.5 Steps Involved in MO-BFFO-TA

Initialize each firefly randomly with either 0 or 1 such that fitness values are greater than 0.0001

Repeat the following steps until Max. Iterations

- Calculate the objective function for all fireflies using eqn.4.
- Replace the weakest Firefly in the population by the solution yielded by Threshold Accepting Algorithm, which is invoked probabilistically.
- For each firefly x_i, we compare its light intensity L_i with the light intensity L_j of every other firefly.
- If $L_i<L_j$, then we move firefly x_i towards x_j in n $_d$imensions using eqns. (2) and (3) and apply sigmoid function to transform real values to binary positions. We do not need to use the bias component in this approach.
- Compute the new value of the objective function for each firefly x_i and update its light intensity.

In MO-BFFO-TA algorithm, we capture the positions and movements of all the fireflies in each of the iterations of MO-BFFO-TA into a database. Then records are ranked and duplicate records are removed. We pick the top M rules as per the objective function. The parameters of MO-BFFO-TA are presented in Table 3.

The following box briefly describes the algorithm:

```
RS = /* initially, Rule Set is empty */
WHILE Iteration Number < Maximum Iterations
        IF rand()<Prob_TA
            Call the Threshold Accepting algorithm to replace theweakest firefly.
            Add all the candidate rules generated in the TA based search to the RS
            Replace the weakest firefly in the population
        ELSE
            Continue MO-BFFO ARM
            Add the rules of all the fireflies in the population of MO-BFFO to the RS in the current
iteration
Rank all the Rules in RS according to objective function
Discard all the duplicate rules
```

Table 3. Parameters chosen for MO-BFFO-TA Rule Miner

Dataset	N	β_0	Υ	Δ	α	MaxIterations	Prob_TA	eps	acc	ThreshTol	Thresh	GI	IO
Book	10	2	2.5	1	0.5	50	0.2	0.01	0.5	0.00018	0.0002	100	25
Food	10	2	2.5	1	0.5	50	0.3	0.01	0.5	0.00018	0.0002	100	25
Grocery	10	2	2.5	1	0.5	50	0.3	0.01	0.5	0.00018	0.0002	100	25
Bank	10	2	2.5	1	0.5	150	0.2	0.01	0.5	1.8E-07	2E-07	200	25
Click stream	10	2	2.5	1	0.5	50	0.3	0.01	0.5	0.000018	0.00002	200	25
Bakery	10	2	2.5	1	0.5	200	0.8	0.01	0.5	1.8E-06	0.000002	200	25

3.6 Steps Involved in MO-BPSO-TA

Initialize each particle randomly with either 0 or 1 such that fitness values are greater than 0.0001

Repeat the following steps until Max. Iterations

- Calculate the fitness for all particles using eqn. 4.
- Replace the weakest particle in the population by the solution yielded by Threshold Accepting Algorithm, which is invoked probabilistically, one’s or twice in hundred iterations
- Update the local best and global best values.
- Update particles position as in the usual PSO, wherein sigmoid function is applied to transform real values to binary positions.

In MO-BPSO-TA algorithm, we capture all the particles positions and movements in each of the iterations of MO-BPSO and TA into a database. Then the solutions are ranked and duplicate ones are removed. We pick the top M rules as per the objective function. The parameters of MO-BPSO-TA are presented in Table 4. The following box briefly describes the algorithm:

```
RS = /* initially, Rule Set is empty */
WHILE Iteration Number < Maximum Iterations
        IF rand()<Prob_TA
            Call the Threshold Accepting algorithm to replace the weakest firefly.
            Add all the candidate rules generated in the TA based search to the RS
            Replace the weakest firefly in the population
        ELSE
            Continue MO-BPSO ARM
            Add the rules of all the particles in the population of MO-BPSO to the RS in the current
iteration
Rank all the Rules in RS according to fitness function
Discard all the duplicate rules
```

Table 4. Parameters chosen for MO-BPSO-TA Rule Miner

Dataset	n	Inertia	C1	C2	MaxIterations	Prob_TA	eps	Acc	Thresh Tol	Thresh	GI	II
Book	30	0.8	2	2	50	0.1	0.01	0.5	0.000018	0.00002	50	5
Food	30	0.8	2	2	50	0.3	0.01	0.5	0.00018	0.0002	50	5
Grocery	30	0.8	2	2	100	0.1	0.01	0.5	0.00018	0.0002	50	5
Bank	50	0.8	2	2	50	0.7	0.01	0.5	1.8E-06	0.000002	50	5
Click stream	30	100.8	2	2	50	0.1	0.01	0.5	8E-09	1E-08	50	5
Bakery	30	0.8	2	2	50	0.1	0.01	0.5	0.000018	0.00002	300	25

GI-Global Iterations; II-Inner Iterations

4 Results and Discussion

We conducted the experiments on a system with Microsoft Windows 7 64-bit Operating System, Intel Core i5 processor, clock speed of 2.53GHz and 4 GB RAM. The proposed algorithms were developed using Java Standard Edition (JDK 1.7) with Eclipse IDE. In this paper, we considered six datasets for demonstrating the effectiveness. The first dataset in our study is Books dataset taken from XLMINER tool (www.solver.com/xlminer-data-mining). It includes 10 book types and 2000 customer records. Another dataset considered here is Food dataset taken from IBM SPSS Modeler (www.ibm.com/software/analytics/spss) tool. This dataset contains 11 types of various food items and 1000 transactional records. The third dataset is the grocery dataset taken from SAS enterprise tool (http://www.sas.com/technologies/ analytics/datamining/miner). This dataset contains 20 grocery products and 1001 transactions. We also analyzed real world dataset from XYZ bank, a commercial bank. It is a sparse dataset consisting of 12191 customers' transactional records and 134 different product and service offerings. The fifth dataset is the Bakery dataset (https://wiki.csc.calpoly.edu/datasets/wiki/ExtendedBakery) which has a list of 40 pastry items and 10 coffee drinks with 1000 transactions. The last dataset is the Anonymous Web Dataset (http://archive.ics.uci.edu/ml/datasets/Anonymous+Microsoft+Web+Data). The data records include the use of www.microsoft.com by 37711 anonymous, randomly-selected users where the dataset contains the lists all the areas of the web site (Vroots) that user visited in a one week timeframe. There are 294 features of website links in this dataset. The results (see Tables 5through 10) obtained are discussed as follows. Table 11 presents the computational times for all the three algorithms.

Table 5. Results of Books dataset

	Supp.	Conf.	Cov.	Comp.	Lev.	Int.	Lift	Conv.	Fitness
MO-BPSO	12.58	76.33	18.5	62.25	5.58	34.6	7.39	6.9	0.001506
MO-BPSO-TA	16.44	68.81	24.36	61.3	6.72	27.94	3.16	2.11	0.001096
MO-BFFO-TA	17	68.45	25.32	64.71	6.35	26.13	2.79	2.01	0.000997

Table 6. Results of Food dataset

	Supp.	Conf.	Cov.	Comp.	Lev.	Int.	Lift	Conv.	Fitness
MO-BPSO	13.25	71.15	20.64	55.16	7.07	25.57	2.37	3.7	0.002033
MO-BPSO-TA	12.19	71.41	19.37	54.47	6.38	23.08	2.37	3.78	0.00156
MO-BFFO-TA	9.5	50.88	18.4	60.64	4.63	16.12	2.66	1.5	0.000723

Table 7. Results of Grocery dataset

	Supp.	Conf.	Cov.	Comp.	Lev.	Int.	Lift	Conv.	Fitness
MO-BPSO	14.64	81.12	18.81	65.12	9.97	45.91	4.03	8.72	0.012162
MO-BPSO-TA	21.84	77.93	29.54	63.74	9.25	36.16	2.5	2.7	0.004953
MO-BFFO-TA	12.56	29.98	4.28	69.56	2.7	13.4	1.41	1.1	0.000265

Table 8. Results of Bank dataset

	Supp.	Conf.	Cov.	Comp.	Lev.	Int.	Lift	Conv.	Fitness
MO-BPSO	2.04	54.25	4.54	65.32	1.77	20.95	12.93	3.01	7.52E-05
MO-BPSO-TA	2.15	53.79	4.35	67.94	1.84	18.41	9.79	2.49	7.26E-05
MO-BFFO-TA	1.11	28.62	4.85	64.71	0.69	6.93	4.07	1.65	5.76E-05

Table 9. Results of Bakery dataset

	Supp.	Conf.	Cov.	Comp.	Lev.	Int.	Lift	Conv.	Fitness
MO-BPSO	4.21	56.1	7.87	68.25	3.66	35.2	9.45	4.47	0.000408
MO-BPSO-TA	4.36	51.63	8.41	69.56	3.75	33.75	8.69	1.95	0.000254
MO-BFFO-TA	3.63	47.06	7.67	66.32	3.1	25.2	7.52	1.78	9.92E-05

Table 10. Results of Click Stream dataset

	Supp.	Conf.	Cov.	Comp.	Lev.	Int.	Lift	Conv.	Fitness
MO-BPSO	4.99	60.54	8.93	66.84	3.37	23.02	6.86	3.24	0.000179
MO-BPSO-TA	5.57	53.88	11.09	65.01	3.48	20.54	5.24	2.11	0.000134
MO-BFFO-TA	2.09	23	13.76	69.56	0.95	5.57	2.78	1.27	1.97E-05

Table 11. Computational times for the algorithms

Data Set	MO-BFFO-TA	MO-BPSO-TA	MO-BPSO
BOOKS	3.205 s	2.22 s	20.595 s
FOOD	1.36 s	1.65 s	11.26 s
GROCERY	42.22 s	4.15 s	24.47 s
XYZ Bank	51.81 s	110.49 s	923.10 s
Clickstream	92.02 s	61.51 s	1567.97 s
Bakery	49.79 s	34.433 s	311.55 s

A) Books Dataset

On Books dataset, we applied MO-MO-BPSO association rule miner with the following parameter settings. We have fixed number of particles as 30, inertia as 0.8, constants c_1 and c_2 as 2 and no of iterations as 50. Later, we applied Hybridized MO-BPSO-TA algorithm. We have chosen same MO-BPSO parameters. Here for TA, we chose inner iterations as 50, outer iterations as 5, accuracy as 0.5, epsilon as 0.1 and TA is called with a probability of 10% of number of iterations. Later, we applied Hybridized MO-BFFO-TA based Association rule miner. We chose the number of fireflies as 10, attractiveness constant (β_0) as 2, gamma (γ) as 2.5 and alpha (α) as 0.5, and TA is called with a probability of 20% of total number of iterations. Here, i.e., with MO-BPSO based association rule miner, it extracted rules with higher values of confidence, Interestingness, Lift, Conviction. MO-BPSO-TA extracted rules with higher levels of leverage. MO-BFFOTA produced higher support, coverage, comprehensibility. We conclude that MO-BPSO outperforms all others, as it extracted rules with many higher measures of strength. The MO-BPSO-TA produced fitness values which are near to that of MO-BPSO. But, when time complexity is critical, MO-BPSOTA is preferred. MO-BFFO-TA is found to produce inferior fitness values.

B) Food Dataset

For food dataset, all the parameters for MO-BPSO, MO-BPSOTA, and MO-BFFOTA are the same as for books dataset. However, the probability of calling TA is increased to 30% for MO-BPSO-TA and MO-BFFO-TA algorithms for increasing the efficiency of the algorithm as useful rules are in the limited neighborhood. MO-BPSO extracted rules with higher support, coverage, leverage and interestingness. We observed that MO-BPSO-TA produced higher confidence, conviction values, while MO-BFFO-TA produced rules with higher comprehensibility, lifts values. Similar to Books dataset, MO-BPSO outperformed all other operators on food dataset.

C) Grocery Dataset

Here too, the parameters for MO-BPSO, MO-BPSOTA, and MO-BFFOTA are same as for books dataset except that the maximum iterations for MO-BPSO is increased to 100 due to nature of the dataset. MO-BPSO produced rules with higher confidence, comprehensibility, leverage, Interestingness, lift and conviction values. However, MO-BPSOTA extracted rules with higher support and coverage values. MO-BFFO-TA is found be inferior for all the measures. Similar to Books dataset, MO-BPSO outperformed all others on grocery dataset.

D) Bank Dataset

For Bank dataset, all the parameters for MO-BPSO are same as for books dataset except that the number of particles is increased to 50 due to data sparseness. In case of MO-BPSO-TA, the number of particles is increased to 50 and the TA is called with a probability of 70% of total number of iterations. The parameters for MO-BFFO-TA algorithm are the same as for Books dataset except that the number of iterations is increased to 150. MO-BFFO-TA extracted rules with high coverage, while MO-BPSO yielded rules with higher confidence, Interestingness, lift and conviction values. MO-BPSO-TA extracted rules with high support, comprehensibility and leverage. Again, MO-BFFOTA is found be inferior. Similar to Books dataset, MO-BPSO outperformed all others.

E) Bakery Dataset

The parameters for MO-BPSO are same as for books dataset, while that of MO-BPSO-TA are same as that of books dataset except that the max inner and outer iterations of TA are fixed at 300 and 25 respectively. MO-BPSO extracted rules with higher confidence, interestingness, lift, conviction values. However, MO-BPSO-TA, extracted rules with high support, coverage, comprehensibility, leverage. Again, MO-BFFO-TA was found to be inferior. Similar to Books dataset, MO-BPSO outperformed all other algorithms.

F) Clickstream Dataset

The parameters for MO-BPSO, MO-BPSO-TA are same as that of books dataset except that the inertia for MO-BPSO is increased to 100.8. The parameters for MO-BFFO-TA are the same as that of food dataset. MO-BPSO extracted rules with higher confidence, interestingness, lift, conviction. MO-BPSO-TA extracted rules with higher support, leverage. MO-BFFOTA extracted rules with high coverage, comprehensibility. Here too, MO-BPSO outperformed all others.

5 Conclusion

We proposed three techniques viz., Multi-objective Binary Particle Swarm Optimization (MO-BPSO), a Multi-objective Binary Firefly optimization and Threshold Accepting (MO-BFFO-TA) and a Multi-objective Binary Particle Swarm optimization and Threshold Accepting (MO-BPSO-TA) to extract association rules from databases by optimizing several rule quality measures objectives. The advantage of proposed methods is that of the user need not specify minimum support and confidence. The MO-BFFO-TA and MO-BPSO-TA Rule Miners could generate all the top 10 rules in just a single run. This is a very significant improvement over MO-BPSO. Overall, MO-BFFO-TA and MO-BPSO-TA also consumed less time compared to a priori, FP-Growth. Further, these algorithms do not generate redundant rules.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD Conference on Management of Data, Washington, D.C., USA, pp. 207–216 (1993)
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Proceedings of the 20th Int. Conference on VLDB Conference, Santiago, Chile, pp. 487–499 (1994)
3. Han, J., Pei, H., Yin, Y.: Mining frequent patterns without candidate generation. In: Conference on the Management of Data, SIGMOD 2000, Dallas, pp. 1–12. ACM Press (2000)
4. Saggar, M., Agrawal, A.K., Lad, A.: Optimization of association rule mining using improved genetic algorithms. In: Proceeding of the IEEE International Conference on Systems Man and Cybernetics, vol. 4, pp. 3725–3729 (2004)
5. Anandhavalli, M., Kumar, S.S., Kumar, A., Ghose, M.K.: Optimized association rule mining using genetic algorithm. In: Advances in Information Mining, pp. 1–4 (2009)
6. Waiswa, P.P.W., Baryamureeba, V.: Extraction of interesting association Rules using genetic algorithms. Int. Journal of Computing and ICT Research 2(1), 26–33 (2008)
7. Ghosh, A., Nath, B.: Multi-Objective rule mining using genetic algorithms. Information Sciences 163, 123–133 (2004)
8. Minaei-Bidgoli, B., Barmaki, R., Nasir, M.: Mining numerical association rules via multi-objective genetic algorithms. Information Sciences 233, 15–24 (2013)
9. Kaya, M., Alhajj, R.: Mining optimized fuzzy association rules using multi-objective genetic algorithm. In: 8th IEEE International Conference on Intelligent Engineering Systems, Cluj-Napoca, Romania, pp. 38–43 (2004)
10. Gupta, M.: Application of weighted particle swarm optimization in association rule mining. International Journal of Computer Science and Informatics 1 (2012) 2231–5292
11. Asadi, A., Afzali, M., Shojaei, A., Sulaimani, S.: New binary PSO based method for finding best thresholds in association rule mining. Applied Soft Computing, 260–264 (2012)
12. Nandhini, M., Janani, M., Sivanandham, S.N.: Association rule mining using swarm intelligence and domain ontology. In: IEEE International Conference on Recent Trends in Information Technology (ICRTIT), Coimbatore, pp. 537–541 (2012)
13. Alatas, B., Akin, E., Karci, A.: MODENAR: Multi-objective differential evolution algorithm for mining numeric association rules. Applied Soft Computing, 646–656 (2008)
14. Kuo, R.J., Chao, C.M., Chiu, Y.T.: Application of particle swarm optimization to association rule mining. Applied Soft Computing 11, 326–336 (2011)

15. Menéndez, H.D., Barrero, D.F., Camacho, D.: A multi-objective genetic graph-based clustering algorithm with memory optimization. In: IEEE Congress on Evolutionary Computation (CEC), pp. 3174–3181 (2013)
16. Menéndez, H., Bello-Orgaz, G., Camacho, D.: Features selection from high-dimensional web data using clustering analysis. In: Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics, pp. 20:1–20:9. ACM, New York (2012)
17. Menéndez, H.D., Camacho, D.: A Multi-Objective Graph-based Genetic Algorithm for Image Segmentation. In: Proceedings of the 2014 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA 2014), pp. 234–241 (2014)
18. Sarath, K.N.V.D., Ravi, V.: Association rule mining using binary particle swarm optimization. Engineering Applications of Artificial Intelligence 26, 1832–1840 (2013)
19. Naveen, N., Ravi, V., Rao, C.R., Sarath, K.N.V.D.: Rule extraction using firefly optimization: Application to banking. In: IEEE International Conference on Industrial Engineering and Engineering Management (2012)
20. Maheshkumar, Y., Ravi, V., Abraham, A.: A particle swarm optimization-threshold accepting hybrid algorithm for unconstrained optimization. Neural Network World, 191–221 (2013)
21. Yang, X.S.: Firefly Algorithm, Stochastic test functions and design optimization. International Journal of Bio-Inspired Computation 2(2), 78–84 (2010)
22. Dueck, G., Scheuer, T.: Threshold Accepting: A general purpose optimization algorithm appearing superior to simulated annealing. Journal of Computational Physics 90(1), 161–175 (1990)
23. Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science (Nagoya, Japan), pp. 39–43. IEEE Service Center, Piscataway (1995)
24. Wur, S.Y., Leu, Y.: An effective Boolean algorithm for mining association rules in large databases. In: Proceedings of the 6th International Conferenceon Database Systems for Advanced Applications, pp. 179–186 (1998)
25. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995)
26. Kennedy, J., Eberhart, R.C.: A discrete binary version of the particle swarm algorithm. In: Proceedings of the Conference on Systems, Man, and Cybernetics, pp. 4104–4109 (1997)
27. Li, Y., Ning, P., Wang, X.S., Jajodia, S.: Discovering calendar-based temporal association rules. Data and Knowledge Engineering 44(2), 193–218 (2003)
28. Extended Bakery Dataset, `https://wiki.csc.calpoly.edu/datasets/wiki/ExtendedBakery`
29. Anonymous Web Dataset, `http://archive.ics.uci.edu/ml/datasets/Anonymous+Microsoft+Web+Data`