Automatic Mining of Numerical Classification Rules with Parliamentary Optimization Algorithm

Soner KIZILOLUK[1], Bilal ALATAS[2]
[1]*Department of Computer Engineering, Tunceli University, Tunceli, Turkey*
[2]*Department of Software Engineering, Firat University, 23119, Elazig, Turkey*
balatas@firat.edu.tr

Abstract—**In recent years, classification rules mining has been one of the most important data mining tasks. In this study, one of the newest social-based metaheuristic methods, Parliamentary Optimization Algorithm (POA), is firstly used for automatically mining of comprehensible and accurate classification rules within datasets which have numerical attributes. Four different numerical datasets have been selected from UCI data warehouse and classification rules of high quality have been obtained. Furthermore, the results obtained from designed POA have been compared with the results obtained from four different popular classification rules mining algorithms used in WEKA. Although POA is very new and no applications in complex data mining problems have been performed, the results seem promising. The used objective function is very flexible and many different objectives can easily be added to. The intervals of the numerical attributes in the rules have been automatically found without any a priori process, as done in other classification rules mining algorithms, which causes the modification of datasets.**

Index Terms—**Classification algorithms, Computational intelligence, Data mining, Heuristic algorithms, Optimization**

I. Introduction

Data mining aims to extract interesting, valid and potentially useful information from large amounts of data [1]. One of the crucial fields of data mining is classification rules mining. Classification is the process of finding rules by analyzing train datasets (i.e., objects whose class label is known) and is used for determining the class of objects whose class label is unknown. Many classification models have been proposed, such as decision trees, neural networks, Bayesian classification, support vector machines and *k*-nearest neighbor [1-3]. Most of these models are black-box based methods. However obtaining the comprehensible and accurate rules within datasets is very important in classification task of data mining. Furthermore, mining classification rules within datasets which have numerical attributes is more complex. The classification rule mining methods used for numerical data perform a kind of discretization as a priori process. In this situation; the problem, the dataset in data mining, is changed. That is why, found classification model is model of the changed dataset. Changing the data set is unreasonable. Adapting the mining algorithm without changing the dataset is more reasonable.

Optimization is selection the best solution of a problem from other alternative solutions. Metaheuristic optimization algorithms are preferred in highly nonlinear and multimodal real-world optimizations with various complex constraints and different conflicting objectives. Some metaheuristic optimization algorithms are inspired by biological, physical, social, and chemical processes [4-7]. Ant colony optimization [8-9] and artificial immunology [10-11] are biology based methods. Simulated annealing [12] and electromagnetism algorithm [13] are physics based methods. Tabu search [14], imperialist competitive algorithm [15], parliamentary optimization algorithm [16], and teaching-learning optimization algorithm [17] are sociology inspired methods. Artificial chemical reaction optimization algorithm [7] is chemistry based method.

In this work, one of the newest social-based metaheuristic methods, Parliamentary Optimization Algorithm (POA), has been used for comprehensible and accurate classification rules mining within numerical data which is a complex problem. Automatically finding the accurate and comprehensible classification rule sets with appropriate intervals of related attributes without a priori process such as discretization using a flexible objective function satisfying different objectives has been easily performed with the newest social based metaheuristic algorithm. This paper is organized as follows. Section 2 describes POA. Section 3 explains how to adapt POA for classification rules mining problem. Experimental and comparative results of POA's classification rules mining performance are presented in Section 4; and Section 5 concludes the paper.

II. Parliamentary Optimization Algorithm

POA is inspired by competition within parties in parliament. Pseudo-code of the POA is shown in Fig. 1 [16, 18-21].

A. Population Initialization and Partitioning

The initial population consists of randomly *M* individuals. After that initialized individuals are partitioned in *N* group and each group has *L* individuals. Best *θ* individuals of each group are considered as candidates and other members are considered as regular members [18-21].

B. Intra-group Competition

After initialization and partitioning, regular members of a group are biased toward candidates. New position of members is calculated as in (1).

This work is supported by the Scientific Research Project Fund of Tunceli University under the project number YLTUB011-14.

Digital Object Identifier 10.4316/AECE.2015.04003

$$p' = p_0 + \pi \left(\frac{\sum_{i=1}^{\theta} (p_i - p_0) f(p_i)}{\sum_{i=1}^{\theta} f(p_i)} \right) \qquad (1)$$

```
❑ Step 1: Population initializing
❑ Step 2: Partitioning population into M groups each with L individuals
      ▪ Step 2.1: Picking $\vartheta$ highly fitted individuals as candidates of each group
❑ Step 3: Intra-group competition
      ▪ Step 3.1: Biasing regular members toward candidates of each group
      ▪ Step 3.2: Reassigning new candidates
      ▪ Step 3.3: Computing power of each group
❑ Step 4: Inter-group cooperation
      ▪ Step 4.1: Picking $\lambda$ most powerful groups and merge them with probability $P_m$
      ▪ Step 4.2: Removing $\gamma$ weak groups with probability $P_d$
❑ Step 5: If terminating condition is not met go to Step 3
❑ Step 6: Returning the best candidate as the solution of the problem
```

Figure 1. Pseudo-code of the POA

Л is a random value between 0.5 and 2, p' is the new position of the regular member, p_0 is the current position of the regular member and p_i is a candidate member and f is the fitness function.

Then power of group is calculated using (2).

$$power_i = \frac{a \times average(Q_i) + b \times average(R_i)}{a+b}; a > b \qquad (2)$$

Q_i and R_i are fitness values of candidates and regular members of the group i respectively. a and b are weighting constants values of which are determined before algorithm execution. The biasing operation is shown in Fig. 2. P0 is regular member, P1, P2, and P3 are candidates and P' is new position of regular member [18].

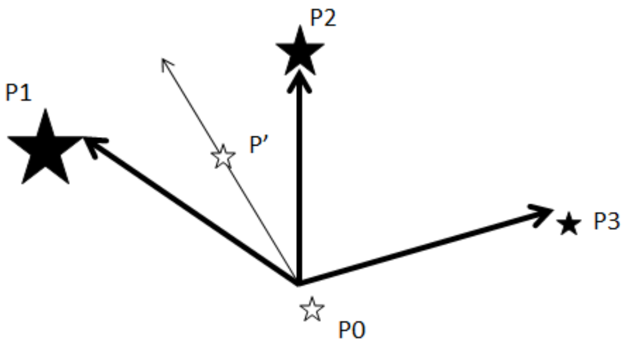


Figure 2. Biasing operation

C. Inter-group Competition

A random number is generated and if it is smaller than P_m, λ most powerful groups are merged into a single group. Fig. 3 illustrates the merging two groups. Like merging, a random number is generated and if it is smaller than P_d, Υ least powerful groups are disappeared [18].

D. Terminating the Algorithm

If maximum number of iterations is reached or there is no increment in fitness values during some successful iterations, then algorithm terminates. At the end of the algorithm, best member of best group is the solution of the optimization problem [18]. Optimization process flowchart of POA is demonstrated in Fig. 4.

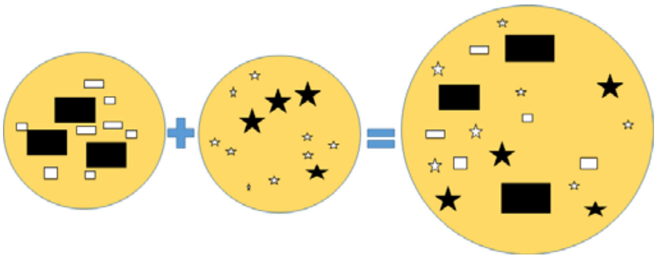


Figure 3. Merging two groups

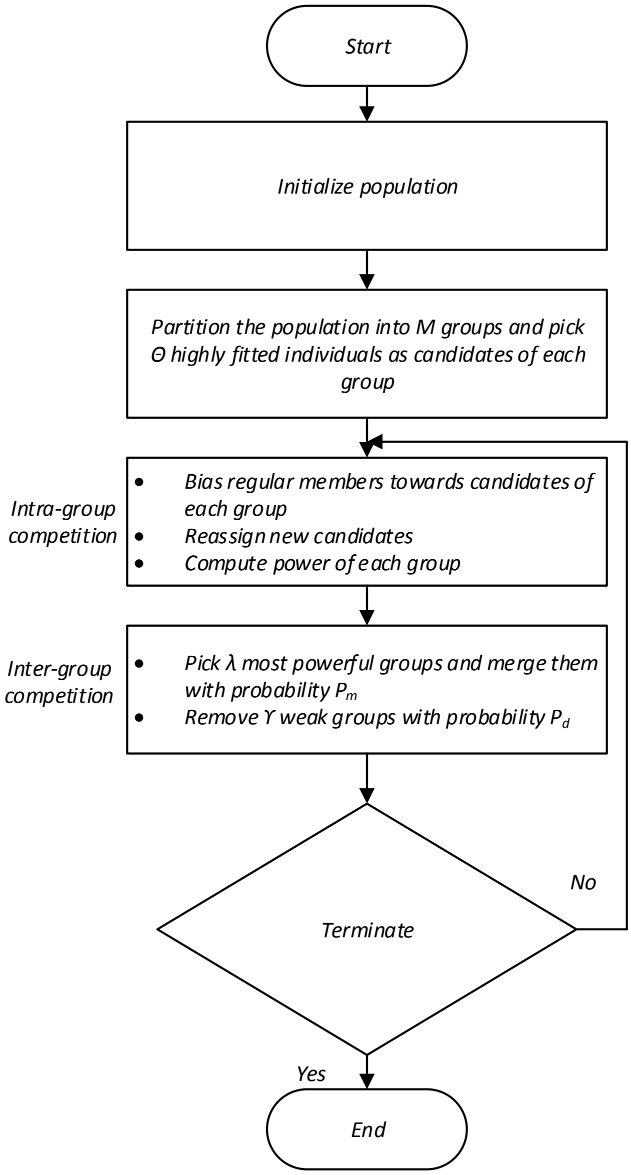

Start
Initialize population
Partition the population into M groups and pick Θ highly fitted individuals as candidates of each group
Intra-group competition
Bias regular members towards candidates of each group
Reassign new candidates
Compute power of each group
Inter-group competition
Pick λ most powerful groups and merge them with probability Pm
Remove ϒ weak groups with probability Pd
Terminate
No
Yes
End

Figure 4. Flowchart of the POA

III. Adapting the POA for Classification Rules Mining

In this study, POA for numerical classification rules mining has been programmed with Visual C#. The detailed adaptation of the algorithm has been described in the subsections.

A. Initializing the Population

Initially, 200 individuals have been initialized taking account of the number of attributes and lower and upper bound of these attributes. Indeed, each individual is a candidate classification rule.

Individuals contain all attributes except "class" attribute in the dataset which is used by algorithm. All attributes have 3 subdivisions. General structure of individuals is shown in Fig. 5.

Att₁			***Att₂***			**…**	***Attₙ***		
F_1	Lb_1	Ub_1	F_2	Lb_2	Ub_2	**…**	F_n	Lb_n	Ub_n

Figure 5. General structure of individuals

In this structure, *F* is flag. If *F* is "1", related attribute is in the antecedent of the candidate rule and if it is "0", related attribute is not included in the antecedent of the candidate rule. *Lb* is lower bound and *Ub* is upper bound of the attribute for possible rule. *F*, *Lb*, and *Ub* are initialized randomly. For understanding the encoding scheme, assume a candidate classification rule created for New Thyroid dataset as shown in Fig. 6.

<table>
<tr><th colspan="3"><i>T3resin</i></th><th colspan="3"><i>Thyroxin</i></th><th colspan="3"><i>Triiodothyronine</i></th></tr>
<tr><td>1</td><td>69</td><td>90</td><td>0</td><td>0.5</td><td>5.5</td><td>0</td><td>0.5</td><td>3.5</td></tr>
<tr><th colspan="6"><i>Thyroidstimulating</i></th><th colspan="3"><i>TSH_value</i></th></tr>
<tr><td colspan="2">0</td><td colspan="2">0.1</td><td colspan="2">11</td><td>1</td><td>0.5</td><td>4.5</td></tr>
</table>

Figure 6. An encoded candidate classification rule for New Thyroid dataset

The candidate solution has five subdivisions due to the New Thyroid dataset's five attributes. Flags which have "1" will form the antecedent of the classification rule and in this candidate solution, "T3resin" and "TSH_value" will be in the antecedent part of the rule. If this solution is a candidate rule for "Class 1", this encoded candidate will be decoded as and represent the rule "**If** (69≤T3resin≤90) and (0.5≤TSH_value≤4.5) **Then** class = 1". This encoded candidate rule will get a fitness function value according to the decoded meaning which shows its quality by means of the important objectives in classification rules mining task.

B. Population Partitioning and Computing Fitness Values

Population is portioned into *M* divisions. Each group has *L* individuals. In this study, value of *M* has been selected as 10 and value of *L* has been selected as 20. However, while algorithm running, groups might earn different number of individuals due to the merge and remove mechanisms in the step of inter-group cooperation. In each group, top $\theta < L/3$ individuals with high fitness have considered as candidates member of each group. Remaining individuals have considered as regular member of group.

Members' fitness values are calculated using (3).

$$fitness = w_1 \times \frac{TP}{TP+FN} \times \frac{TN}{TN+FP} - w_2 \times comprehensibility + \quad (3)$$
$$w_3 \times \frac{TP}{TP+FP} \pm w_4 \times intervalrate$$

TP = Number of true positive instances (the actual class of the test instance is positive and the POA correctly predicts the class as positive)

FP = Number of false positive instances (the actual class of the test instance is negative but the POA incorrectly predicts the class as positive)

FN = Number of false negative instances (the actual class of the test instance is positive but the POA incorrectly predicts the class as negative)

TN = Number of true negative instances (the actual class of the test instance is negative and the POA correctly predicts the class as negative)

w_1, w_2, w_3, and w_4 are weighting factors.

comprehensibility is calculated using (4).

$$comprehensibility = \frac{number\ of\ attributes\ in\ the\ left\ side\ of\ the\ rule\text{-}1}{number\ of\ all\ attributes} \quad (4)$$

intervalrate is calculated as shown in (5)

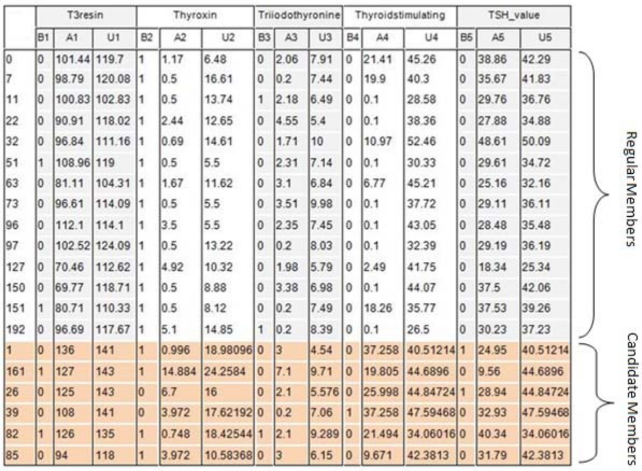
$$intervalrate = \sum_{i=1}^{n} \frac{Ub_i - Lb_i}{Att_{i\max} - Att_{i\min}} \quad (5)$$

In equation (5), *Ub* is upper bound and *Lb* is lower bound of the attributes in the candidate rule. Att_{imax} and Att_{imin} are maximum and minimum values of attributes in the dataset respectively. In the fitness function, there is no obligation of using the *interval rate*. In this study, interval rate has been used only in the Diabetes dataset.

C. Intra-group Competition

Members' new positions are calculated using equation 1. Then, groups' powers are computed as shown in equation 2. In this study, *a* and *b* values in the equation 2 have been selected as 9 and 3, respectively. After biasing operator, regular members may have higher fitness values than those of candidate members. In these situations, members are interchanged.

As an example, a group with 20 members fitness values of which have been ordered in ascending order for New Thyroid dataset has been demonstrated in Fig. 7. Best 6 members are candidate members. *F* (Flag), *Lb* (Lower bound), and *Ub* (Upper bound) values of regular members will be biased to the *F*, *Lb*, and *Ub* values of the candidate members and this biasing will be repeated in each iteration.

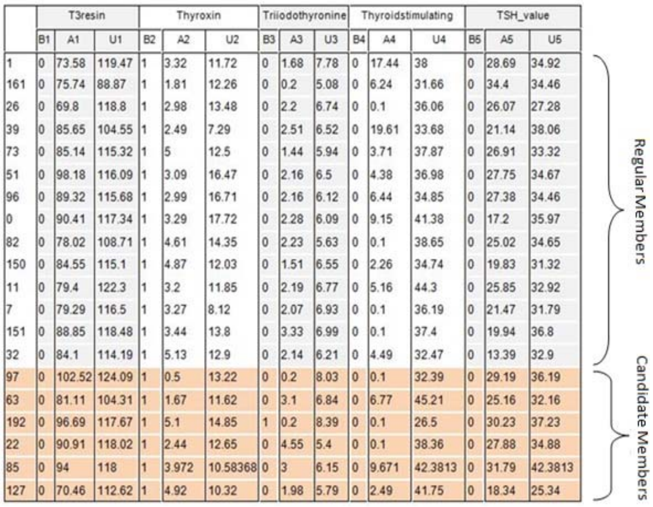


	T3resin			Thyroxin			Triiodothyronine			Thyroidstimulating			TSH_value		
	B1	A1	U1	B2	A2	U2	B3	A3	U3	B4	A4	U4	B5	A5	U5
0	0	101.44	119.7	1	1.17	6.48	0	2.06	7.91	0	21.41	45.26	0	38.86	42.29
7	0	98.79	120.08	1	0.5	16.61	0	0.2	7.44	0	19.9	40.3	0	35.67	41.83
11	0	100.83	102.83	1	0.5	13.74	1	2.18	6.49	0	0.1	28.58	0	29.76	36.76
22	0	90.91	118.02	1	2.44	12.65	0	4.55	5.4	0	0.1	38.36	0	27.88	34.88
32	0	96.84	111.16	1	0.69	14.61	0	1.71	10	0	10.97	52.46	0	48.61	50.09
51	1	108.96	119	1	0.5	5.5	0	2.31	7.14	0	0.1	30.33	0	29.61	34.72
63	0	81.11	104.31	1	1.67	11.62	0	3.1	6.84	0	6.77	45.21	0	25.16	32.16
73	0	96.61	114.09	1	0.5	5.5	0	3.51	9.98	0	0.1	37.72	0	29.11	36.11
96	0	112.1	114.1	1	3.5	5.5	0	2.35	7.45	0	0.1	43.05	0	28.48	35.48
97	0	102.52	124.09	1	0.5	13.22	0	0.2	8.03	0	0.1	32.39	0	29.19	36.19
127	0	70.46	112.62	1	4.92	10.32	0	1.98	5.79	0	2.49	41.75	0	18.34	25.34
150	0	69.77	118.71	1	0.5	8.88	0	3.38	6.98	0	0.1	44.07	0	37.5	42.06
151	1	80.71	110.33	1	0.5	8.12	0	0.2	7.49	0	18.26	35.77	0	37.53	39.26
192	0	96.69	117.67	1	5.1	14.85	1	0.2	8.39	0	0.1	26.5	0	30.23	37.23
1	0	136	141	1	0.996	18.98096	0	3	4.54	0	37.258	40.51214	1	24.95	40.51214
161	1	127	143	1	14.884	24.2584	0	7.1	9.71	0	19.805	44.6896	0	9.56	44.6896
26	0	125	143	0	6.7	16	0	2.1	5.576	0	25.998	44.84724	1	28.94	44.84724
39	0	108	141	0	3.972	17.62192	0	0.2	7.06	1	37.258	47.59468	0	32.93	47.59468
82	1	126	135	1	0.748	18.42544	1	2.1	9.289	0	21.494	34.06016	0	40.34	34.06016
85	0	94	118	1	3.972	10.58368	0	3	6.15	0	9.671	42.3813	0	31.79	42.3813

Figure 7. Initial group and selected candidates

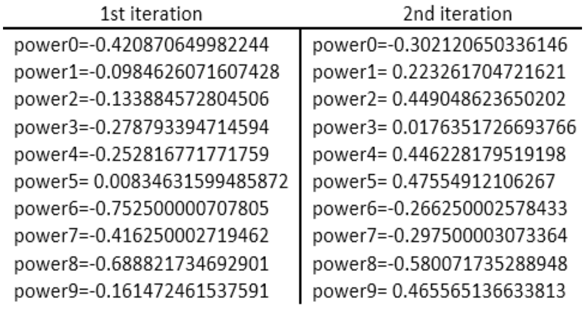
The values of the same group after one iteration have been demonstrated in Fig. 8. *F*, *Lb*, and *Ub* values of regular members have changed. Furthermore, 5 regular members have gotten bigger fitness values than those of other candidate members except from candidate member no 85. That is why, members are interchanged.

After biasing, powers of each group are computed again. Fig. 9 shows the powers of the groups after first and second iteration. After one iteration, the powers of the groups have been increased. This means that, fitness values of the members of the groups have been increased and rules of good quality will be obtained.



	T3resin			Thyroxin			Triiodothyronine			Thyroidstimulating			TSH_value		
	B1	A1	U1	B2	A2	U2	B3	A3	U3	B4	A4	U4	B5	A5	U5
1	0	73.58	119.47	1	3.32	11.72	0	1.68	7.78	0	17.44	38	0	28.69	34.92
161	0	75.74	88.87	1	1.81	12.26	0	0.2	5.08	0	6.24	31.66	0	34.4	34.46
26	0	69.8	118.8	1	2.98	13.48	0	2.2	6.74	0	0.1	36.06	0	26.07	27.28
39	0	85.65	104.55	1	2.49	7.29	0	2.51	6.52	0	19.61	33.68	0	21.14	38.06
73	0	85.14	115.32	1	5	12.5	0	1.44	5.94	0	3.71	37.87	0	26.91	33.32
51	0	98.18	116.09	1	3.09	16.47	0	2.16	6.5	0	4.38	36.98	0	27.75	34.67
96	0	89.32	115.68	1	2.99	16.71	0	2.16	6.12	0	6.44	34.85	0	27.38	34.46
0	0	90.41	117.34	1	3.29	17.72	0	2.28	6.09	0	9.15	41.38	0	17.2	35.97
82	0	78.02	108.71	1	4.61	14.35	0	2.23	5.63	0	0.1	38.65	0	25.02	34.65
150	0	84.55	115.1	1	4.87	12.03	0	1.51	6.55	0	2.26	34.74	0	19.83	31.32
11	0	79.4	122.3	1	3.2	11.85	0	2.19	6.77	0	5.16	44.3	0	25.85	32.92
7	0	79.29	116.5	1	3.27	8.12	0	2.07	6.93	0	0.1	36.19	0	21.47	31.79
151	0	88.85	118.48	1	3.44	13.8	0	3.33	6.99	0	0.1	37.4	0	19.94	36.8
32	0	84.1	114.19	1	5.13	12.9	0	2.14	6.21	0	4.49	32.47	0	13.39	32.9
97	0	102.52	124.09	1	0.5	13.22	0	0.2	8.03	0	0.1	32.39	0	29.19	36.19
63	0	81.11	104.31	1	1.67	11.62	0	3.1	6.84	0	6.77	45.21	0	25.16	32.16
192	0	96.69	117.67	1	5.1	14.85	1	0.2	8.39	0	0.1	26.5	0	30.23	37.23
22	0	90.91	118.02	1	2.44	12.65	0	4.55	5.4	0	0.1	38.36	0	27.88	34.88
85	0	94	118	1	3.972	10.58368	0	3	6.15	0	9.671	42.3813	0	31.79	42.3813
127	0	70.46	112.62	1	4.92	10.32	0	1.98	5.79	0	2.49	41.75	0	18.34	25.34

Figure 8. Group after one iteration and selected candidates



1st iteration	2nd iteration
power0=-0.420870649982244	power0=-0.302120650336146
power1=-0.0984626071607428	power1= 0.223261704721621
power2=-0.133884572804506	power2= 0.449048623650202
power3=-0.278793394714594	power3= 0.0176351726693766
power4=-0.252816771771759	power4= 0.446228179519198
power5= 0.00834631599485872	power5= 0.47554912106267
power6=-0.752500000707805	power6=-0.266250002578433
power7=-0.416250002719462	power7=-0.297500003073364
power8=-0.688821734692901	power8=-0.580071735288948
power9=-0.161472461537591	power9= 0.465565136633813

Figure 9. Powers of the groups

D. Inter-group Competition

In this study, a random number between 0 and 100 has been generated and P_m, λ, P_d, Υ values have been selected as 3, 2, 1, and 2, respectively.

E. Terminating Condition

In this study, algorithm terminates if number of iterations is reached 100. At the end of the algorithm, most powerful group's best member is a classification rule of the dataset.

IV. Experiments

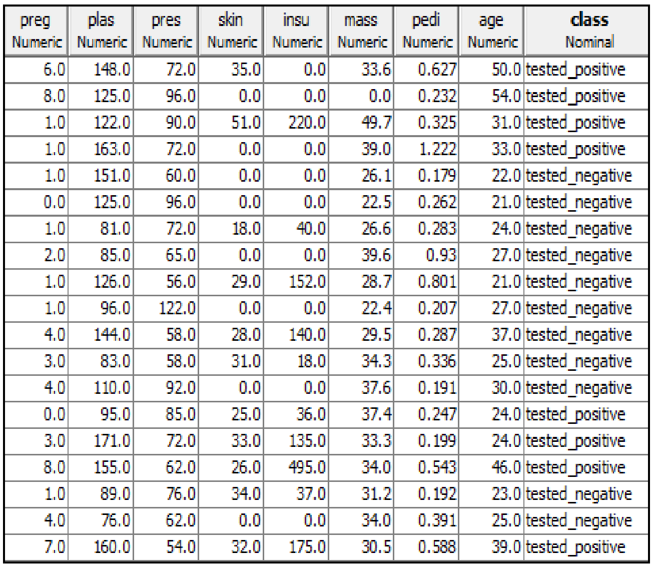
Four datasets (Pima Indians Diabetes, Ecoli, BUPA Liver Disorders, and New Thyroid) from the UCI machine learning repository have been used. In these datasets, all attributes have real or integer values.

A. Datasets

Pima Indians Diabetes dataset contains 768 instances, each with 8 attributes. These 8 attributes and their minimum and maximum values are shown in Table I. Class attribute of dataset has 2 different values ("tested_positive" or "tested_negative"). 500 of the instances belong to "tested_negative" and 268 of the instances belong to "tested_positive" class. A section from Pima Indians Diabetes dataset is shown in Fig. 10.

TABLE I. ATTRIBUTES OF PIMA INDIANS DIABETES

Attribute	*Min. value*	*Max. value*
Preg	0	17
Plas	0	199
Pres	0	122
Skin	0	99
Insu	0	846
Mass	0	67.1
Pedi	0.078	2.42
Age	21	81



preg Numeric	plas Numeric	pres Numeric	skin Numeric	insu Numeric	mass Numeric	pedi Numeric	age Numeric	class Nominal
6.0	148.0	72.0	35.0	0.0	33.6	0.627	50.0	tested_positive
8.0	125.0	96.0	0.0	0.0	0.0	0.232	54.0	tested_positive
1.0	122.0	90.0	51.0	220.0	49.7	0.325	31.0	tested_positive
1.0	163.0	72.0	0.0	0.0	39.0	1.222	33.0	tested_positive
1.0	151.0	60.0	0.0	0.0	26.1	0.179	22.0	tested_negative
0.0	125.0	96.0	0.0	0.0	22.5	0.262	21.0	tested_negative
1.0	81.0	72.0	18.0	40.0	26.6	0.283	24.0	tested_negative
2.0	85.0	65.0	0.0	0.0	39.6	0.93	27.0	tested_negative
1.0	126.0	56.0	29.0	152.0	28.7	0.801	21.0	tested_negative
1.0	96.0	122.0	0.0	0.0	22.4	0.207	27.0	tested_negative
4.0	144.0	58.0	28.0	140.0	29.5	0.287	37.0	tested_negative
3.0	83.0	58.0	31.0	18.0	34.3	0.336	25.0	tested_negative
4.0	110.0	92.0	0.0	0.0	37.6	0.191	30.0	tested_negative
0.0	95.0	85.0	25.0	36.0	37.4	0.247	24.0	tested_positive
3.0	171.0	72.0	33.0	135.0	33.3	0.199	24.0	tested_positive
8.0	155.0	62.0	26.0	495.0	34.0	0.543	46.0	tested_positive
1.0	89.0	76.0	34.0	37.0	31.2	0.192	23.0	tested_negative
4.0	76.0	62.0	0.0	0.0	34.0	0.391	25.0	tested_negative
7.0	160.0	54.0	32.0	175.0	30.5	0.588	39.0	tested_positive

Figure 10. A section from Pima Indians Diabetes dataset

Ecoli dataset has 336 instances, each with 7 attributes. Table II shows that attributes and their minimum and maximum values. Class attribute of dataset has 8 different values (cp, im, pp, imU, om, omL, imL, and imS). Class attributes and number of attributes belong to these classes are shown in Table III. A section from Ecoli dataset is also shown in Fig. 11.

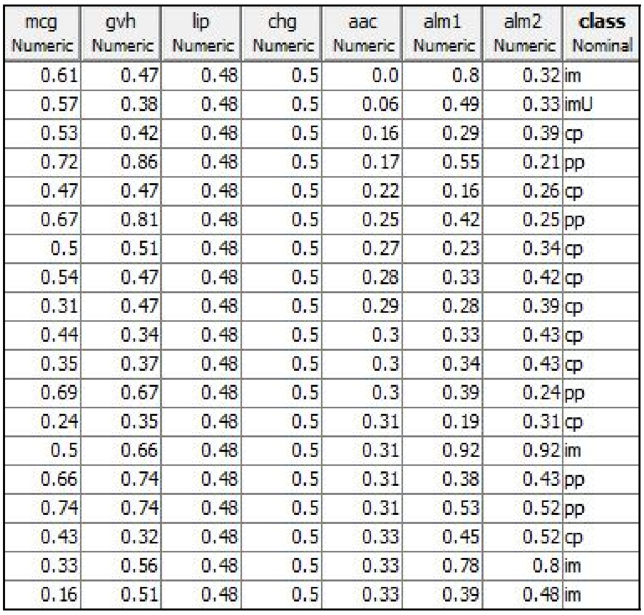
TABLE II. ATTRIBUTES OF ECOLI

Attribute	*Min. value*	*Max. value*
Mcg	0	0.89
gvh	0.16	1
lip	0.48	1
chg	0.5	1
aac	0	0.88
alm1	0.03	1
alm2	0	0.99

TABLE III. CLASS ATTRIBUTE OF ECOLI

Class	cp	im	pp	imU	om	omL	imL	imS
No of instances	143	77	52	35	20	5	2	2

BUPA Liver Disorders dataset contains 345 instances, each with 6 attributes. Attributes and their minimum and maximum values are shown in Table IV. Class attribute of BUPA dataset has 2 different values ("1" or "2"). 145 of the instances belong to class "1" and 200 of the instances belong to class "2". A section from Bupa dataset is also shown in Fig. 12.

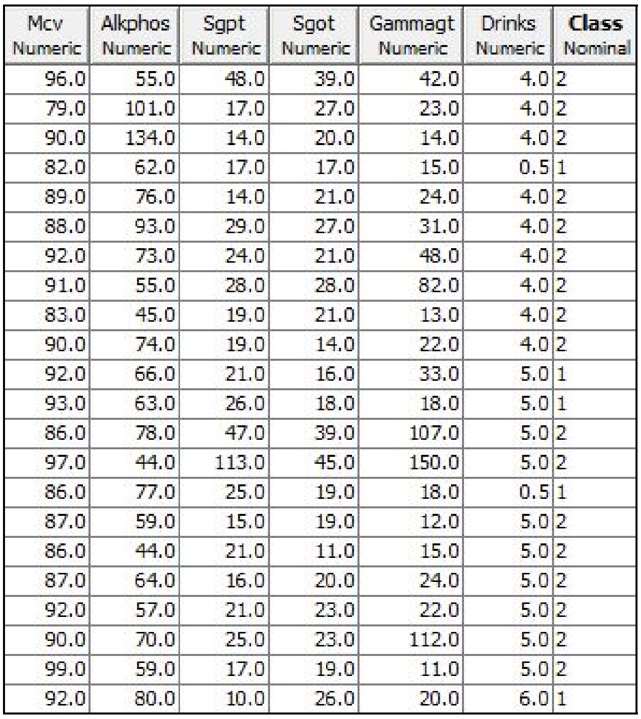


mcg Numeric	gvh Numeric	lip Numeric	chg Numeric	aac Numeric	alm1 Numeric	alm2 Numeric	class Nominal
0.61	0.47	0.48	0.5	0.0	0.8	0.32	im
0.57	0.38	0.48	0.5	0.06	0.49	0.33	imU
0.53	0.42	0.48	0.5	0.16	0.29	0.39	cp
0.72	0.86	0.48	0.5	0.17	0.55	0.21	pp
0.47	0.47	0.48	0.5	0.22	0.16	0.26	cp
0.67	0.81	0.48	0.5	0.25	0.42	0.25	pp
0.5	0.51	0.48	0.5	0.27	0.23	0.34	cp
0.54	0.47	0.48	0.5	0.28	0.33	0.42	cp
0.31	0.47	0.48	0.5	0.29	0.28	0.39	cp
0.44	0.34	0.48	0.5	0.3	0.33	0.43	cp
0.35	0.37	0.48	0.5	0.3	0.34	0.43	cp
0.69	0.67	0.48	0.5	0.3	0.39	0.24	pp
0.24	0.35	0.48	0.5	0.31	0.19	0.31	cp
0.5	0.66	0.48	0.5	0.31	0.92	0.92	im
0.66	0.74	0.48	0.5	0.31	0.38	0.43	pp
0.74	0.74	0.48	0.5	0.31	0.53	0.52	pp
0.43	0.32	0.48	0.5	0.33	0.45	0.52	cp
0.33	0.56	0.48	0.5	0.33	0.78	0.8	im
0.16	0.51	0.48	0.5	0.33	0.39	0.48	im

Figure 11. A section from Ecoli dataset

TABLE IV. ATTRIBUTES OF BUPA DATASET

Attribute	*Min. value*	*Max. value*
Mcv	65	103
Alkphos	23	138
Sgpt	4	155
Sgot	5	82
Gammagt	5	297
Drinks	0	20



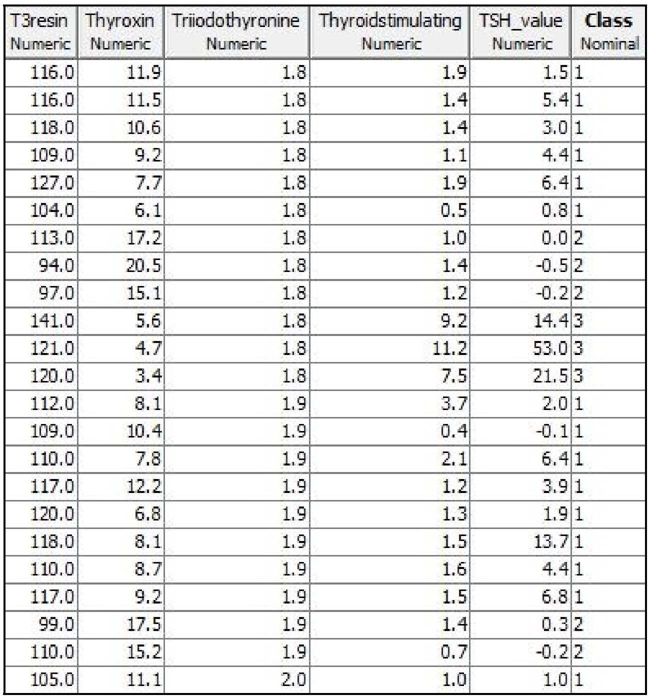
Mcv Numeric	Alkphos Numeric	Sgpt Numeric	Sgot Numeric	Gammagt Numeric	Drinks Numeric	Class Nominal
96.0	55.0	48.0	39.0	42.0	4.0	2
79.0	101.0	17.0	27.0	23.0	4.0	2
90.0	134.0	14.0	20.0	14.0	4.0	2
82.0	62.0	17.0	17.0	15.0	0.5	1
89.0	76.0	14.0	21.0	24.0	4.0	2
88.0	93.0	29.0	27.0	31.0	4.0	2
92.0	73.0	24.0	21.0	48.0	4.0	2
91.0	55.0	28.0	28.0	82.0	4.0	2
83.0	45.0	19.0	21.0	13.0	4.0	2
90.0	74.0	19.0	14.0	22.0	4.0	2
92.0	66.0	21.0	16.0	33.0	5.0	1
93.0	63.0	26.0	18.0	18.0	5.0	1
86.0	78.0	47.0	39.0	107.0	5.0	2
97.0	44.0	113.0	45.0	150.0	5.0	2
86.0	77.0	25.0	19.0	18.0	0.5	1
87.0	59.0	15.0	19.0	12.0	5.0	2
86.0	44.0	21.0	11.0	15.0	5.0	2
87.0	64.0	16.0	20.0	24.0	5.0	2
92.0	57.0	21.0	23.0	22.0	5.0	2
90.0	70.0	25.0	23.0	112.0	5.0	2
99.0	59.0	17.0	19.0	11.0	5.0	2
92.0	80.0	10.0	26.0	20.0	6.0	1

Figure 12. A section from Bupa dataset

Thyroid Disease (New Thyroid) Dataset have 215 instances, each with 5 attributes. These 5 attributes and their minimum and maximum values are shown in Table V. A section from Thyroid Disease dataset is also shown in Fig. 13.

TABLE V. ATTRIBUTES OF NEW THYROID DATASET

Attribute	*Min. value*	*Max. value*
T3resin	65	144
Thyroxin	0.5	25.3
Triiodothyronine	0.2	10
Thyroidstimulating	0.1	56.4
TSH_value	-0.7	56.3



T3resin Numeric	Thyroxin Numeric	Triiodothyronine Numeric	Thyroidstimulating Numeric	TSH_value Numeric	Class Nominal
116.0	11.9	1.8	1.9	1.5	1
116.0	11.5	1.8	1.4	5.4	1
118.0	10.6	1.8	1.4	3.0	1
109.0	9.2	1.8	1.1	4.4	1
127.0	7.7	1.8	1.9	6.4	1
104.0	6.1	1.8	0.5	0.8	1
113.0	17.2	1.8	1.0	0.0	2
94.0	20.5	1.8	1.4	-0.5	2
97.0	15.1	1.8	1.2	-0.2	2
141.0	5.6	1.8	9.2	14.4	3
121.0	4.7	1.8	11.2	53.0	3
120.0	3.4	1.8	7.5	21.5	3
112.0	8.1	1.9	3.7	2.0	1
109.0	10.4	1.9	0.4	-0.1	1
110.0	7.8	1.9	2.1	6.4	1
117.0	12.2	1.9	1.2	3.9	1
120.0	6.8	1.9	1.3	1.9	1
118.0	8.1	1.9	1.5	13.7	1
110.0	8.7	1.9	1.6	4.4	1
117.0	9.2	1.9	1.5	6.8	1
99.0	17.5	1.9	1.4	0.3	2
110.0	15.2	1.9	0.7	-0.2	2
105.0	11.1	2.0	1.0	1.0	1

Figure 13. A section from Thyroid Disease (New Thyroid) dataset

B. Experimental Results

All instances of datasets have been used as training set; also the mined rules have been tested with these instances. The obtained results for the all datasets have been compared with 4 algorithms in WEKA. These algorithms are Jrip, Ridor, Part, and One-R.

Results for Pima Indians Diabetes Dataset

Weighting values used for Pima Indians Diabetes dataset are shown in Table VI.

TABLE VI. WEIGHTING VALUES USED FOR PIMA INDIANS DIABETES DATASET

w_1	w_2	w_3	w_4
0.28	0.20	0.40	0.12

The results obtained by POA and WEKA are shown in Table VII and Table VIII.

TABLE VII. RESULTS OF POA FOR PIMA INDIANS DIABETES DATASET

	Number of rules	*Number of correctly classified instances / Total number of instances*	*Accuracy (%)*
1st run	13	611/768	79.55
2nd run	13	609/768	79.29
3rd run	6	609/768	79.29
4th run	7	600/768	78.12
Avg.			**79.06**

Mined rules for diabetes dataset classified the data with

79.06% average accuracy. In the above tables, the results clearly show that POA's results are similar with Jrip's and Ridor's results. Besides, results obtained by POA are better than the results obtained by One-R and worse than the Part's results.

TABLE VIII. RESULTS OF WEKA FOR PIMA INDIANS DIABETES DATASET

Algorithm	*Number of rules*	*Number of correctly classified instances / Total number of instances*	*Accuracy (%)*
Jrip	4	609/768	79.29
Ridor	4	605/768	78.77
Part	13	624/768	81.25
One-R	10	586/768	76.30

Mined 6 rules in the 3rd run of POA are shown in Table IX.

TABLE IX. MINED RULES BY POA IN THE 3RD RUN

Rule no	*Rules*
1	**If** (40≤plas≤127.99) **Then** class = tested_negative.
2	**If** (55.81≤plas≤145.51) and (21.85≤mass≤28.27) **Then** class = tested_negative.
3	**If** (0≤preg≤1.26) and (63.66≤plas≤152.33) and (69.98≤pres≤86.5) and (18.08≤mass≤45.09) **Then** class = tested_negative.
4	**If** (38.5≤pres≤110) and (18.05≤mass≤29.97) **Then** class = tested_negative.
5	**If** (40≤plas≤148.03) and (72.06≤pres≤101.26) and (18.98≤mass≤53.1) and (0≤pedi≤0.39) **Then** class = tested_negative.
6	**If** others **Then** class = tested_positive.

Results for Ecoli Dataset

Weighting values used for Ecoli dataset are shown in Table X.

TABLE X. WEIGHTING VALUES USED FOR ECOLI DATASET

w_1	w_2	w_3
0.20	0.20	0.60

POA's and WEKA's results are shown in Table XI and Table XII.

TABLE XI. RESULTS OF POA FOR ECOLI DATASET

	Number of rules	*Number of correctly classified instances / Total number of instances*	*Accuracy (%)*
1st run	12	285/336	84.82
2nd run	22	290/336	86.70
3rd run	12	274/336	81.54
4th run	18	281/336	83.63
Avg.			**84.17**

TABLE XII. RESULTS OF WEKA FOR ECOLI DATASET

Algorithm	*Number of rules*	*Number of correctly classified instances / Total number of instances*	*Accuracy (%)*
Jrip	10	305/336	90.77
Ridor	46	297/336	88.39
Part	13	308/336	91.66
One-R	7	232/336	69.04

Obtained rules for Ecoli dataset classified the data with 84.17% average accuracy. Except One-R, the other algorithms (Jrip, Ridor, and Part) in WEKA have better results than the POA's result.

Mined 12 rules in the 1st run of POA are shown in Table XIII.

TABLE XIII. MINED RULES BY POA IN THE 1ST RUN

Rule no	*Rules*
1	**If** (0.14≤mcg≤0.55) and (0.22≤alm2≤0.61) **Then** class = cp.
2	**If** (0.16≤gvh≤0.42) and (0.1≤aac≤0.88) and (0≤alm2≤0.6) **Then** class = cp.
3	**If** (0.06≤ mcg≤0.61) and (0.18≤gvh≤0.87) and (0.05≤aac≤0.88) and (0.6≤alm2≤0.97) **Then** class = im.
4	**If** (0.57≤gvh≤0.7) and (0.18≤aac≤0.88) and (0.55≤alm1≤0.89) and (0.22≤alm2≤0.93) **Then** class = im.
5	**If** (0.28≤mcg≤0.71) and (0.68≤aac≤0.88) and (0.63≤alm1≤0.84) and (0≤alm2≤0.78) **Then** class = im.
6	**If** (0.63≤mcg≤0.66) and (0.52≤gvh≤0.81) and (0.39≤aac≤0.88) and (0.57≤alm1≤0.8) and (0.22≤alm2≤0.97) **Then** class = im.
7	**If** (0.36≤gvh≤0.56) and (0.3≤aac≤0.88) and (0.91≤alm2≤0.99) **Then** class = im.
8	**If** (0.25≤aac≤0.88) and (0.57≤alm≤0.81) and (0.57≤alm2≤0.89) **Then** class = imU.
9	**If** (0.46≤gvh≤0.48) and (0.1≤aac≤0.68) and (0.17≤alm2≤0.99) **Then** class = om.
10	**If** (0.57≤gvh≤0.89) and (0.23≤aac≤0.88) and (0.29≤alm1≤0.63) and (0.16≤alm2≤0.94) **Then** class = pp.
11	**If** (0.48≤mcg≤0.78) and (lip=0.48) and (0.48≤alm1≤0.69) and (0.35≤alm2≤0.95) **Then** class = pp.
12	**If** (0.31≤mcg≤0.72) and (0.37≤gvh≤1) and (0.21≤aac≤0.88) and (0.46≤alm1≤0.68) **Then** class = omL.

Results for the BUPA Liver Disorders Dataset

Weighting values used for BUPA Liver Disorders dataset are shown in Table XIV.

TABLE XIV. WEIGHTING VALUES USED FOR BUPA DATASET

w_1	w_2	w_3
0.20	0.20	0.60

The results obtained by POA and WEKA are shown in Table XV and Table XVI.

TABLE XV. RESULTS OF POA FOR BUPA DATASET

	Number of rules	*Number of correctly classified instances / Total number of instances*	*Accuracy (%)*
1st run	16	276/345	80.00
2nd run	14	276/345	80.00
3rd run	15	273/345	79.13
4th run	15	270/345	78.26
Avg.			**79.34**

TABLE XVI. RESULTS OF WEKA FOR BUPA DATASET

Algorithm	*Number of rules*	*Number of correctly classified instances / Total number of instances*	*Accuracy (%)*
Jrip	5	270/345	78.26
Ridor	3	246/345	71.30
Part	15	297/345	86.08
One-R	14	235/345	68.11

Running the POA for BUPA dataset, mined rules have classified the data with 79.34% average accuracy. In WEKA, result of Part is successful than the results obtained from POA whereas the other WEKA algorithms Jrip, Ridor, and One-R have poorer results than that of POA. Mined 14 rules in the 2nd run of POA are shown in Table XVII.

TABLE XVII. MINED RULES BY POA IN THE 2ND RUN

Rule no	*Rules*
1	**If** (45.6≤sgot≤56.25) **Then** class = 2
2	**If** (70.4≤mcv≤89.47) and (41.96≤gammagt≤62.2) **Then** class = 2.
3	**If** (3.54≤drinks≤5.54) **Then** class = 2.
4	**If** (82.44≤mcv≤85.67) and (11.62≤sgot≤43.18) **Then** class = 2.
5	**If** (24.41≤sgot≤47.26) and (36.94≤gammagt≤55.1) **Then** class = 2.
6	**If** (90.84≤mcv≤96.85) and (1.91≤drinks≤2) **Then** class = 2.
7	**If** (4≤sgpt≤71.15) and (9.99≤drinks≤12.18) **Then** class = 2.
8	**If** (79.99≤mcv≤90.52) and (5.41≤drinks≤13.86) **Then** class = 2.
9	**If** (83.73≤mcv≤95.4) and (26.12≤alkphos≤51.63) and (21.81≤sgot≤48.6) **Then** class = 2.
10	**If** (55.46≤alkphos≤57.08) **Then** class = 2.
11	**If** (11.79≤sgpt≤16.64) **Then** class = 2.
12	**If** (51.98≤alkphos≤94.05) and (35.52≤gammagt≤40.69) **Then** class = 2.
13	**If** (46.18≤alkphos≤84.27) and (64.14≤gammagt≤87.64) **Then** class = 2.
14	**If** others **Then** class = 1.

Results for the Thyroid Disease (New Thyroid) Dataset

Weighting values are shown in Table XVIII.

TABLE XVIII. WEIGHTING VALUES USED FOR NEW THYROID DATASET

w_1	w_2	w_3
0.20	0.20	0.60

The results obtained by POA and WEKA are shown in Table XIX and Table XX.

TABLE XIX. RESULTS OF POA FOR NEW THYROID DATASET

	Number of rules	*Number of correctly classified instances / Total number of instances*	*Accuracy (%)*
1st run	7	211/215	98.13
2nd run	5	205/215	95.34
3rd run	8	201/215	93.48
4th run	5	206/215	95.81
Avg.			**95.69**

TABLE XX. RESULTS OF WEKA FOR NEW THYROID DATASET

Algorithm	*Number of rules*	*Number of correctly classified instances / Total number of instances*	*Accuracy (%)*
Jrip	4	209/215	97.20
Ridor	7	206/215	95.81
Part	4	213/215	99.06
One-R	3	198/215	92.09

TABLE XXI. MINED RULES BY POA IN THE 1ST RUN

Rule no	*Rules*
1	**If** (6.56≤Thyroxin≤11.9) **Then** class = 1.
2	**If** (106.33≤T3resin≤144) and (11.14≤Thyroxin≤14.28) **Then** class = 1.
3	**If** (97.51≤T3resin≤101.78) and (0.6≤TSH_value≤36.96) **Then** class = 1.
4	**If** (1.39≤TSH_value≤54.75) and (6.81≤Thyroxin≤16.1) **Then** class = 1.
5	**If** (103.21≤T3resin≤112.99) and (3.77≤Thyroxin≤11.44) **Then** class = 1.
6	**If** (8.73≤Thyroxin≤25.3) **Then** class = 2.
7	**If** others **Then** class = 3.

All algorithms have high accuracy rules for New Thyroid dataset. The accuracy ratio of POA, Jrip, Ridor, Part, and One-R algorithms are 95.69%, 97.20%, 95.81%, 99.06%, and 92.09% respectively. These results present that POA has approximately same result with the Jrip's and the Ridor's results, better than One-R's result, and worse than Part's results.

Mined 7 rules in the 1st run of POA are shown in Table XXI.

V. CONCLUSION

Automating the classification rules mining task of data mining has required the investigation of many methods, techniques and algorithms. Most of these models are black-box based methods. POA has been firstly used in complex numerical classification rules mining task of data mining. Although POA is very new and no applications in complex data mining problems have been performed, the results seem promising. POA, proposed as numerical classification rules miner in this study, has been designed in such a way that it can obtain the comprehensible and accurate rules set within datasets. By this way, the induced classification model can be interpreted and the obtained rules set can be efficiently used. That is why, the classification model is not black-box such as used methods in the related literature.

Another advantages of the POA designed in this study is flexible nature of the objective function. Any other different objectives such as interestingness, surprisingness, and etc. can be easily added.

Another superiority of the method proposed in this study is finding the numerical intervals of the attributes in the time of rules mining without any a priori data process. This is one of the very best superiority of the designed POA over other classification rules mining algorithms which perform a kind of discretization or fuzzification as a priori process. In this situation; the problem, the dataset in data mining, is changed. That is why, found classification model is model of the changed dataset. Changing the data set is unreasonable. Adapting the mining algorithm without changing the dataset is more reasonable. In this study, POA has been efficiently designed as a rules miner finding the attribute intervals simultaneously.

One of the other advantages of the designed POA is its global search with a population of candidate solutions. It does not start and keep up the search with a single candidate solution for the rules. However, most data mining algorithms usually performs a local search. Furthermore, POA is successful in coping with attribute interaction problem by not selecting one attribute at a time and not evaluating a partially constructed candidate rule. This is also a superiority of the proposed method over other classification algorithms.

Comparing results from WEKA and this study indicate that obtained rules for Diabetes dataset classified the data with 79.06% average accuracy. However, WEKA classified same data with different algorithms which have distinct accuracy such as; Jrip 79.29%, Ridor 78.77%, Part 81.25%, and One-R 76.30%. These results clearly show that POA's result is competitive with the Jrip's result and the Ridor's result. Besides, results obtained from POA are better than the results obtained from One-R and worse than the results obtained from Part algorithm.

Obtained rules for Ecoli dataset classified the data with

84.17% average accuracy. Processing same data in the WEKA, except One-R (69.04%), the others algorithms (Jrip 90.77%, Ridor 88.39%, and Part 91.66%) have better results than the POA's result.

Running the POA for BUPA dataset, mined rules have classified the data with 79.34% average accuracy. In WEKA, result for Part (86.08%) is successful than the results obtained from this study whereas the other WEKA algorithms (Jrip 78.26%, Ridor 71.30%, and One-R 68.11%) have poor results.

All algorithms have mined rules with high accuracy for Thyroid dataset. The accuracy ratio of POA, Jrip, Ridor, Part, and One-R algorithms are 95.69%, 97.20%, 95.81%, 99.06%, and 92.09% respectively. These results present that, POA has approximately same results with the Jrip and the Ridor; better results than One-R, and worse results than Part.

All in all, POA is an effective method for automatic numerical classification rules mining. Although there is no much more optimization on POA, it has mined rules with high accuracy and comprehensibility. POA can also be effectively used for complex numerical association rules mining, sequential patterns mining, and clustering rules mining tasks of data mining. Parallel and distributed version of POA with optimized parameters may be a future work. Generalization of POA for multi-objective optimization problems may also be one of the further works.

References

[1] J. Han and M. Kamber, Data Mining: Concepts and Techniques, Second Edition, Morgan Kaufmann, San Francisco, ch. 1, 2006.

[2] P. Kosina, J. Gama, “Very fast decision rules for classification in data streams”, Data Mining and Knowledge Discovery, vol. 29, no. 1, pp. 168-202, 2015. [Online]. Available: http://dx.doi.org/10.1007/s10618-013-0340-z

[3] M. Hacibeyoglu, A. Arslan, S. Kahramanli, “A hybrid method for fast finding the reduct with the best classification accuracy”, Advances in Electrical and Computer Engineering, vol. 13, no. 4, pp. 57-64, 2013. [Online]. Available: http://dx.doi.org/10.4316/AECE.2013.04010

[4] E. D. Ülker, A. Haydar, “Comparing the robustness of evolutionary algorithms on the basis of benchmark functions”, Advances in Electrical and Computer Engineering, vol. 13, no. 2, pp. 59-64, 2013. [Online]. Available: http://dx.doi.org/10.4316/AECE.2013.02010

[5] R. Ngamtawee, P. Wardkein, “Simplified genetic algorithm: simplify and improve RGA for parameter optimizations”, Advances in Electrical and Computer Engineering, vol. 14, no. 4, pp. 55-64, 2014. [Online]. Available: http://dx.doi.org/10.4316/AECE.2014.04009

[6] B. Alatas, “A novel chemistry based metaheuristic optimization method for mining of classification rules”, Expert Systems with Applications, vol. 39, no. 12, pp. 11080–11088, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2012.03.066

[7] S. Bechikh, A. Chaabani, L.B. Said, “An efficient chemical reaction optimization algorithm for multiobjective optimization”, IEEE Transactions on Cybernetics, vol. 45, no. 10, pp. 2051–2064, 2015. [Online]. Available: http://dx.doi.org/10.1109/TCYB.2014.2363878

[8] M. Dorigo, V. Maniezzo, A. Colorni, “The ant system: Optimization by a colony of cooperating agents”, IEEE Transactions on Systems, Man, and Cybernetics, Part B., vol. 26, no. 1, pp. 29–41, 2002. [Online]. Available: http://dx.doi.org/10.1109/3477.484436

[9] B. Alatas, E. Akin, “FCACO: Fuzzy Classification Rules Mining Algorithm with Ant Colony Optimization”, Lecture Notes in Computer Science, vol. 3612, pp. 787-797, 2005. [Online]. Available: http://dx.doi.org/10.1007/11539902_97

[10] L. De Castro and J. Timmis, Artificial Immune Systems: A New Computational Intelligence Approach, Springer, ch. 3, 2002.

[11] B. Alatas, E. Akin, “Mining Fuzzy Classification Rules Using an Artificial Immune System with Boosting”, Lecture Notes in Computer Science, vol. 3631, pp. 283-293, 2005. [Online]. Available: http://dx.doi.org/10.1007/11547686_21

[12] S. Kirkpatrick, C. D. Gerlatt, M. P. Vecchi, “Optimization by simulated annealing”, Science, vol. 220, no. 4598, pp. 671–680, 1983. [Online]. Available: http://dx.doi.org/10.1126/science.220.4598.671

[13] I. Birbil and S. Fang, “An electromagnetism-like mechanism for global optimization”, Journal of Global Optimization, vol. 25, no. 3, pp. 263–282, 2003. [Online]. Available: http://dx.doi.org/10.1023/A:1022452626305

[14] F. Glover, “Tabu search-part I”, ORSA Journal on Computing, vol. 1, no. 3, pp. 190–206, 1989. [Online]. Available: http://dx.doi.org/10.1287/ijoc.1.3.190

[15] E. Atashpaz-Gargari and C. Lucas, 2007, “Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition”, IEEE Congress on Evolutionary Computation, 2007, pp. 4661–4667. [Online]. Available: http://dx.doi.org/10.1109/CEC.2007.4425083

[16] A. Borji, “A new global optimization algorithm inspired by parliamentary political competitions”, Lecture Notes in Computer Science, vol. 4827, pp. 61–71, 2007. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-76631-5_7

[17] R. V. Rao, V. J. Savsani, D. P. Vakharia, “Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems”, Information Sciences, vol. 183, no. 1, pp. 1–15, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.ins.2011.08.006

[18] A. Borji and M. Hamidi, “A new approach to global optimization motivated by parliamentary political competitions”, Int. Journal of Innovative Computing, Information and Control, vol. 5, no. 6, pp. 1643–1653, 2009.

[19] F. Altunbey, B. Alatas, “Overlapping community detection in social networks using parliamentary optimization algorithm”, International Journal of Computer Networks and Applications, vol. 2, no. 1, 12-19, 2015.

[20] L. de-Marcos, A. Garcia-Cabot, E. Garcia-Lopez, J. Medina, “Parliamentary optimization to build personalized learning paths: Case study in web engineering curriculum”, The International Journal of Engineering Education, vol. 31, no. 4, 2015.

[21] L. de-Marcos, A. García, E. García, J. J. Martínez, J. A. Gutiérrez, R. Barchino, J. M. Gutiérrez, J. R. Hilera, S. Otón, “An adaptation of the parliamentary metaheuristic for permutation constraint satisfaction”, IEEE Congress on Evolutionary Computation (CEC), pp.1-8, 2010. [Online]. Available: http://dx.doi.org/10.1109/CEC.2010.5585915