*Journal of **AI** and **D**ata **M**ining*
Vol 8, No 1, 2020, 25-37.

DOI: 10.22044/JADM.2019.8507.1989

An Evolutionary Multi-objective Discretization based on Normalized Cut

M. H. Tahan* and M. Ghasemzadeh

Electrical and Computer Engineering Department, Yazd University, Yazd, Iran.

Received 01 June 2019; Revised 20 October 2019; Accepted 29 November 2019

**Corresponding author: m.ghasemzadeh@yazd.ac.ir (M. Ghasemzadeh).*

Abstract

Learning models and the related results depend on the quality of the input data. If the raw data is not properly cleaned and structured, the results obtained tend to be incorrect. Therefore, discretization, as one of the pre-processing techniques, plays an important role in learning processes. The most important challenge in the discretization process is to reduce the number of features' values. This operation should be applied in a way that the relationships between the features are maintained, and the accuracy of the classification algorithms would increase. In this paper, a new evolutionary multi-objective algorithm is presented. The proposed algorithm uses three objective functions in order to achieve a high-quality discretization. The first and second objectives minimize the number of the selected cut points and classification error, respectively. The third objective introduces a new criterion called the normalized cut, which uses the relationships between their features' values to maintain the nature of the data. The performance of the proposed algorithm is tested using 20 benchmark datasets. According to the comparisons and the results of the non-parametric statistical tests, the proposed algorithm has been found to have a better performance than the other major existing methods.

Keywords: *Discretization, Multi-objective, Evolutionary, Normalized Cut, Multivariate.*

1. Introduction and literature review

In knowledge discovery, data pre-processing is known as one of the most important steps. Since almost all the data mining processes require high-quality and structured data, pre-processing of the raw data is an essential step in most analytical problems [1, 2]. In this regard, data reduction is one of the major tasks accomplished in pre-processing. The data reduction techniques are often used to reduce the size of the original data and to clean some of the errors that could be present in the data [3].
Data discretization is a data reduction technique that converts complex continuous features into a finite set of discrete intervals. Lately, the data science community has paid a great amount of attention to data discretization [4].
In practice, some of the data mining algorithms only work with discrete features, while in the real world, most problems deal with continuous values. Also some data mining algorithms may produce low-quality results when they directly deal with the continuous data.
In these cases, feature discretization approaches play an important role in converting the continuous features to the discrete ones. In addition, it eliminates the noise and the missing values as well as the unusable and meaningless values. Discretization can also reduce and simplify the data; this usually leads to a faster learning and more accurate, more compact, and shorter results [5-7].
There are various features available to categorize discretization methods including supervised versus unsupervised, splitting versus merging, univariate versus multivariate, etc. [6]. Supervised methods such as MDLP [8], EMD [2], MEMOD [6], and EMDID [7] consider class information, while unsupervised methods do not consider class information and emphasize on the nature of the data. Splitting methods start with one interval and select the best cut point in each step, while merging

methods start with all the candidate cut points, and in each step, the closest intervals merge. Univariate methods discrete each feature individually, while multivariate methods consider the relationship between the features. Most of the previous methods such as CAIM [9], MDLP [8], and Modified-Chi2 [10] are univariate. Since these methods do not consider communication between features, important information is lost and cannot obtain global optima. As a result, multivariate methods such as EMD [5], MEMOD [6], EMDID [7], and GraphS/GraphM [11] have been proposed.

Besides, there are various techniques available for discretization including binning, statistical, information, evolutionary, and hybrid. Evolutionary algorithms (EAs) are one of the most important and successful techniques that can be useful for solving the discretization problem [5]. Data discretization can be solved as an optimization problem so that problem solutions can be coded through the binary presentation. The categorization of some of the evolutionary discretization algorithms is shown in table 1.

Table 1. A categorization of evolutionary discretization algorithms in the literature.

Discretization Algorithm	EAs	No. of objectives	Objectives
GAFD [12]	GA	Single objective	Minimize the classification error
ISCADABPSO [13]	PSO	Single objective	Minimize the classification error, minimize the number of cut points
ECPSD [14]	GA	Single objective	Minimize the data consistency, minimize the number of cut points
EMD [5]	GA	Single objective	Minimize the classification error, minimize the number of cut points
MultiCAIM [15]	NSGA-II	Multi-Objective	Minimize the classification error, minimize the loss of class-attribute interdependency
MEMOD [6]	NSGA-II	Multi-Objective	Minimize the classification error, minimize the number of cut points, minimize the total frequency of selected cut points
EMDID [7]	NSGA-II	Multi-Objective	Minimize the area under ROC curve, minimize the number of cut points, minimize the total frequency of selected cut points

One of the best-known evolutionary discretization methods is EMD [5]. The fitness function of EMD is based on the minimization of the classification error and the number of cut points, while the selected cut points may damage the nature of the data [11]. Also EAs look for global optimization but standard implementation often converges to a local optimum. In addition, it is not possible to consider several conflicting objectives simultaneously [16]. These approaches can, on average, produce satisfactory results but each one of the objectives might be unacceptable separately [17-19]. Thus multi-objective evolutionary discretization algorithms such as MEMOD [6] and EMDID [7] have been introduced to solve this problem. These algorithms solve the discretization problem with the multi-objective method, and they introduce a novel criterion, namely the total frequency of the selected cut points. Using low-frequency values as the cut points, the information loss can be avoided. Regardless of the search algorithm used for discretization, these algorithms evaluate the potential solutions only in terms of the prediction accuracy, and do not focus on the nature of the data. In these algorithms, the objective functions are based on the minimization of the classification error and the number of selected cut points. While the cut points may be caused, the nature of the data and hidden patterns between the data will inevitably be lost [20]. Recently, a discretization algorithm based on the graph clustering has been presented, which uses the similarity measures and the class of instances to examine the similarity between the data values [20]. It mainly focuses on the relationship between the features and the nature of the data but ignores the relationship between the features and classes.

In this paper, an evolutionary multi-objective algorithm based on non-dominated sorting genetic algorithm-III (NSGA-III) is proposed, which uses three objective functions including the number of selected cut points, classification error, and normalized cut. The proposed algorithm uses the normalized cut as an active limit to determine the cut points. This objective function selects every cut point between the intervals by preserving information about the hidden patterns between the data when the data has a high similarity, which helps to increase the purity of the intervals.

The structure of the rest of the paper is as what follows. In the second section, the proposed algorithm is described. The results and evaluation of the tests are presented in the third section. Conclusions and the future works are expressed in the last section.

2. Proposed algorithm

In this section, we introduce an evolutionary multi-objective method for the discretization problem. In the proposed algorithm, a new criterion called a normalized cut is considered in order to evaluate the quality of discretization. This new criterion helps maintain the structure and nature of the data. This approach is such that the set of data points is considered as the chromosome genes of the evolutionary algorithm. Then the cut points are obtained using NSGA-III based on three objective functions including the number of selected cut points, the classification accuracy, and the similarity between the features' values. This algorithm contains several solutions that the user can choose from the obtained solutions based on his/her needs. The steps involved in the proposed algorithm are as follow:

1) Determination of the initial cut points;
2) Creation of the affinity matrix (AF);
3) Application of the NSGA-III algorithm with three objectives: (1) number of selected cut points (2) classification error (3) normalized cut;
4) Creation of the discretization scheme;
5) Conversion of the continuous data to the discrete forms.

These main steps are elaborated in the following.

2.1. Determination of initial cut points

In order to get the initial cut points, the continuous feature A is first arranged in an ascending order. Suppose that $Dom(A)$ represents the domain of the feature A and $Val_A(s)$ indicates the value of the feature in the instance $s \in S$. If there is a pair of instances $u, v \in S$ that have different classes so that $Val_A(u) < Val_A(v)$ and there is no other instance $w \in S$ so that $Val_A(u) < Val_A(w) < Val_A(v)$, the mean of the values u and v is considered in the initial set of cut points.

2.2. Creation of affinity matrix

Affinity matrix is a matrix $n \times n$ that shows the similarity between the pairs of data points. The values for this matrix are between 0 and 1, which represents the similarity between the pairs. Before calculating the values of the matrix, the values of the data are rescaled (normalized) between 0 and 1 using the min-max method in order to give all features the same treatment [11, 20].

The similarity is calculated by (1), which is the weighted sum of the similarity between the data values and similarity between class labels of data pairs. The value in this equation is between zero and one, where zero means that only the similarity between the data values is considered and one means that only the similarity between the class labels is considered.

$$sim(u,v) = (\alpha) simC(u,v) + (1-\alpha) simP(u,v) \quad (1)$$

In order to calculate $SimP(u,v)$, the cosine similarity criterion is used. This criterion calculates the cosine between the angles between two vertices or instances. The cosine similarity between the instances is obtained using (2).

$$simP_C(u,v) = \frac{\sum_{i=1}^{d} u_i v_i}{\sqrt{\sum_{i=1}^{d} u_i^2}\sqrt{\sum_{i=1}^{d} v_i^2}} \quad (2)$$

2.3. Application of NSGA-III algorithm

The non-dominated sorting genetic algorithm-III is a genetic algorithm that simultaneously optimizes several objectives using the non-dominated sorting technique [21]. This algorithm uses a selective operator based on the reference points to explore the solution space and maintain diversity. The steps of the NSGA-III algorithm are as follow:

1) Encoding the chromosome;
2) Initialization of the population (randomly);
3) Calculation of all the reference points;
4) Application of non-dominated sorting;
5) Application of cross-over and selection operators;
6) Re-application of non-dominated sorting;
7) Normalization of members of the population;
8) Assigning points to the reference points;
9) niche preservation;
10) Maintenance of the elite solutions for the next step;
11) Repeating the algorithm until the end condition is satisfied.

In the following, the main steps are discussed in further details.

2.3.1. Encode chromosome

In the evolutionary algorithm, it is first necessary to encode the problem as a chromosome. In the proposed algorithm, the length of the chromosome was considered as the number of the initial cut points. Each chromosome contains the values of zero and one. One means that a cut point is selected, and zero otherwise. At first, the initial chromosomes are randomly assigned.

2.3.2. Objective functions

The objective functions used in the proposed algorithm include the number of selected cut points, classification error, and normalized cut. The algorithm tries to minimize the number of cut points, classification error, and normalized cut criteria. Each one of the objective functions is explained further in the following.

- **Number of selected cut points**

This objective function attempts to reduce the number of cut points. As the number of cut points in the discretization is reduced, the discretized dataset will be simpler and more compact, and it will be easier for the user to understand it. In this algorithm, the total number of chromosome genes that is equal to one is considered as the objective function. This is shown by (3).

$$f_1(S) = Number\ of\ \ selected\ cut\ point \tag{3}$$

- **Classification error**

One of the benefits of discretization is to improve the classification accuracy. The second objective considered in the proposed algorithm is to reduce the classification error. Two classifiers were used to measure the classification accuracy including C4.5 and Naïve Bayes (NB). The average errors obtained for these two classes is considered as the total classification error; this is shown by (4). The use of the average error of the two classifiers ensures that the proposed algorithm is not fit on a particular classifier. With this design, we can obtain more effective discretization schemes.

$$f_2(S) = \frac{error_{C4.5} + error_{NB}}{2} \tag{4}$$

- **Normalized cut**

In order to maintain the nature of the data in this algorithm, a new criterion called a normalized cut is used. To increase the quality of the discretization, the values in each interval should be as close as possible to each other, and the values at different intervals vary as much as possible.

For this purpose, the idea of the graph clustering is used. This algorithm initially creates an affinity matrix, and by constructing this matrix, the problem is actually converted to a graph, whose interconnected components in the graph form a cluster (values in one interval). In fact, in this graph, the edges whose elements are in a cluster are weighted more; conversely, the edges whose elements are not in a cluster are weighed less. Therefore, in order to calculate the third objective function in the proposed algorithm, we must use the graph clustering quality evaluation techniques. One of these techniques is the normalized cut that has been used to solve the graph clustering problems (Equation 3). In (3), C_i is cluster i, and, respectively, $\omega(C_i, C_i)$ and $\omega(C_i, \overline{C_i})$ are the sum of weighted edges of the intra-cluster and inter-cluster edges.

$$NCut(\pi_k) = \sum_{i=1}^{k} \frac{\omega(C_i, \bar{C}_i)}{\omega(C_i, C_i) + \omega(C_i, \bar{C}_i)} \tag{5}$$

In order to calculate the third objective function, first, AF for each feature is arranged ascending. All rows and columns are reordered according to one feature. Since ordering is difficult in terms of programming, an identification number (ID) is allocated to each instance, which associates with a row/column number of AF. Then the reorder is only featured with its ID. Finally, we consider the total normalized cut obtained for each feature as the final value of the third objective function. This is represented as (6).

$$f_3(S) = \sum_{i=1}^{Number\ of\ attributes} NCut_i(\pi_k) \tag{6}$$

2.4. Creation of discretization scheme

The discretization intervals are determined based on the set of selected cut points, and the datasets are discretized based on the discretization scheme.

2.5. Conversion of continuous data to discrete forms

Any continuous value of the feature is converted into a discrete value based on the discretization scheme obtained from the previous step. In this way, for each interval obtained, the values in which the interval that is replaced with the name is assigned to that interval.

3. Experimental results and discussion

The proposed algorithm was implemented in Python 3.7. All experiments were conducted on an Intel(R) Core(TM) CPU@2.20 GHz and 8 GB RAM. In this section, the performance evaluation of the proposed algorithm was performed using a variety of datasets. In order to evaluate the performance of the proposed algorithm, 20 benchmark datasets were used. Table 2 shows a summary of the datasets used in the experiments. For each dataset, the number of instances, number of features, and number of classes are shown. The datasets were partitioned using the 10-fold cross-validation method. All datasets are available in the UCI Machine Learning Database [22].

The performance of the proposed algorithm was compared with the well-known algorithms such as CAIM [9], MDLP [8], Mod-Chi2 [10], EMD [5], MEMOD [6], GraphS [11], and GraphM [11]. MDLP, EMD, and MEMOD achieved a good trade-off between the accuracy and the number of selected cut points. Considering the trade-off accuracy and simplicity, one of these methods is a good option. The supervised and unsupervised, direct and incremental, and statistical/information evaluation are characteristics of the best algorithms in terms of the performance. Besides, CAIM, modified-Chi2, EMD, and MEMOD obtained high performances among all types of classifiers [7]. The proposed algorithm was also compared with GraphS and GraphM since these were the new discretizers that were shown to work well on different datasets. The values assigned to the parameters of algorithms were effective in the process of accessing an algorithm to the appropriate and desirable responses. Table 3 shows the parameters used in different algorithms. The parameters are based on the cases the authors of each algorithm have suggested in the article.

Table 2. Properties of datasets.

	Datasets	No. of instances	No. of features	No. of classes
1	**abalone**	4,174	8	28
2	**appendicitis**	106	7	2
3	**balance**	625	4	3
4	**bupa**	345	6	2
5	**contraceptive**	1,473	9	3
6	**glass**	214	9	7
7	**haberman**	306	3	2
8	**iris**	150	4	3
9	**penbased**	10,992	16	10
10	**phoneme**	5,472	5	2
11	**pima**	768	8	2
12	**saheart**	462	9	2
13	**satimage**	6,435	36	7
14	**sonar**	208	60	2
15	**tae**	151	5	3
16	**transfusion**	748	5	2
17	**vehicle**	846	18	4
18	**vowel**	990	13	11
19	**wine**	178	13	3
20	**yeast**	1,484	8	10

3.1. Comparison of performance in terms of classification accuracy and number of cut points

In order to compare the performance of the algorithms, the accuracy of the classification and preserve the nature of data, the proposed algorithm in some datasets such as pima, saheart, vehicle, and wine chooses more cut points. Tables 5, 6, and 7 the number of cut points for the discretization schemes derived from the eight algorithms were compared with 20 datasets in the following. The mean classification accuracy derived from the C4.5 and Naïve Bayes and SVM classifiers was considered as the classification accuracy of the algorithms for comparison.

Table 3. Parameters of discretizers.

Algorithm	Parameters
EMD	Population size = 50 Iterations = 10000 Weight factor (α) = 0.7 Reduction percentage = 0.5 Reduction rate = 0.1
MEMOD	Population size = 50, Iterations = 100 Reduction percentage = 0.5 Reduction rate = 0.1 Selection = binary tournament Cross-over = uniform crossover Cross-over probability = 0.6 Mutation probability = 0.4
GraphS/ GraphM	Weight-determination (α) = 0.2, significant improvement percentage **(β)** = 1.01

Proposed Algorithm	Population size = 50, Iterations = 10000,
	Cross-over = binary cross-over, Simulated,
	mutation = Polynomial, Mutation,
	Weight-determination (α) = 0.2

Figures 1, 2, and 3 show the Pareto front obtained from the proposed algorithm based on the only two objectives of the classification error (sum of classification error of NB and C4.5 classifier) and the number of selected cut points. Since the objective based on the normalized cut is hidden in other objectives, except for some datasets such as penbased, phoneme, sonar, tae, and vehicle, the proposed algorithm has been able to overcome other algorithms in the literature. For these datasets, the proposed algorithm has been able to overcome most of the algorithms. As shown in figures 1, 2, and 3, the proposed multi-objective algorithm has the capability to find several solutions that a user can choose from one solution based on the fact that the number of cut points is more important to him/her or the classification accuracy.

Among the solutions obtained by the algorithm, a solution was chosen; the results obtained can be seen in tables 4, 5, 6, and 7. The results were used for the non-parametric statistical test. According to the results in table 4, the proposed algorithm was able to obtain a relatively small number of cut points. The mean number of cut points by the proposed algorithm is more than MEMOD. This is due to the nature of the proposed algorithm. To show the results of the classification accuracy of the algorithms on NB, C4.5, and SVM classifier, respectively. Table 5 shows that the proposed algorithm was able to obtain a better classification accuracy in the twelve datasets for the NB classifier. Tables 6 and 7 show that the best mean classification accuracies for the C4.5 and SVM classifiers belong to the proposed algorithm.

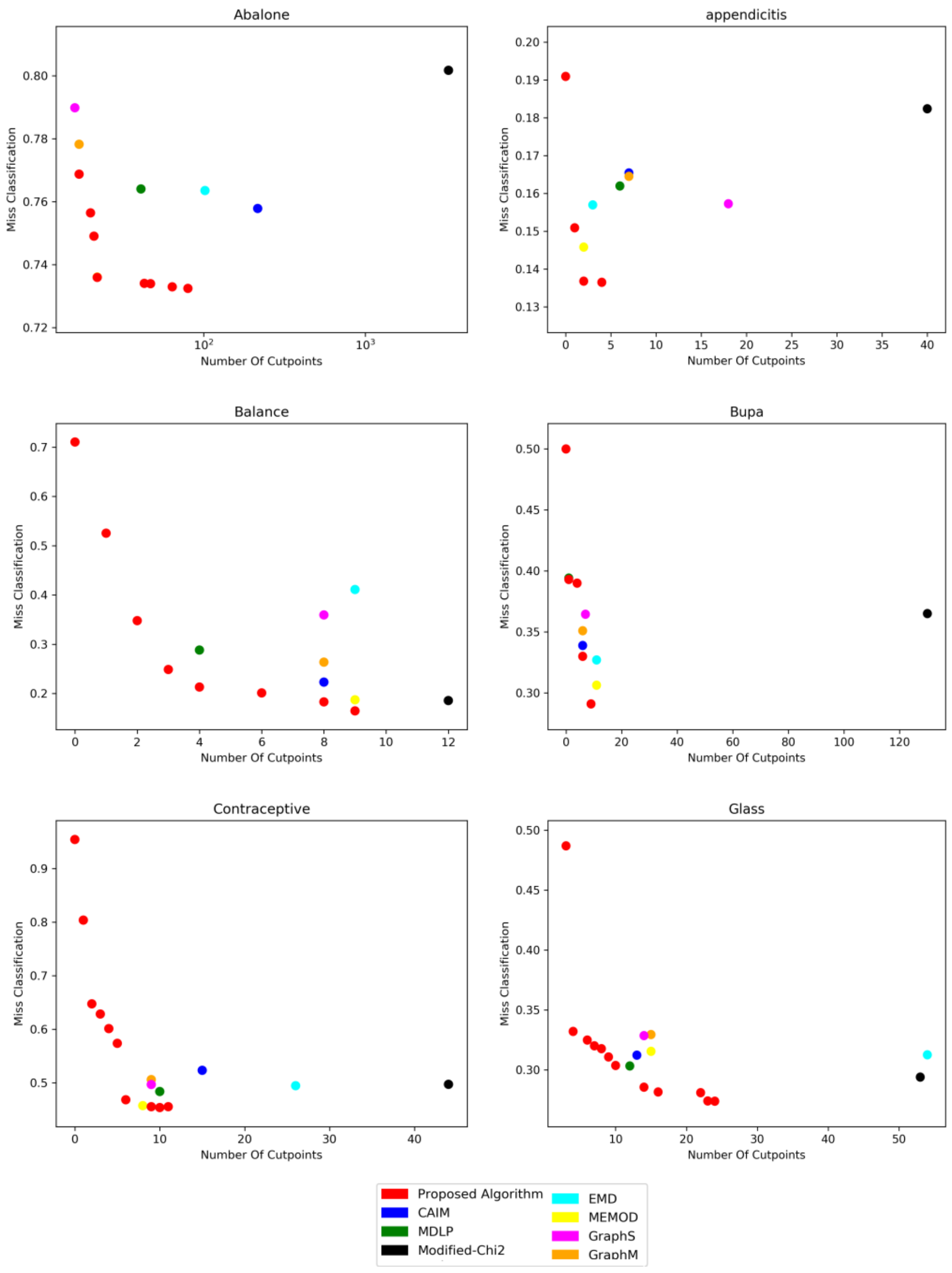


Figure 1. Projection of non-dominated solutions obtained by the proposed algorithm for the considered datasets (Part-1).

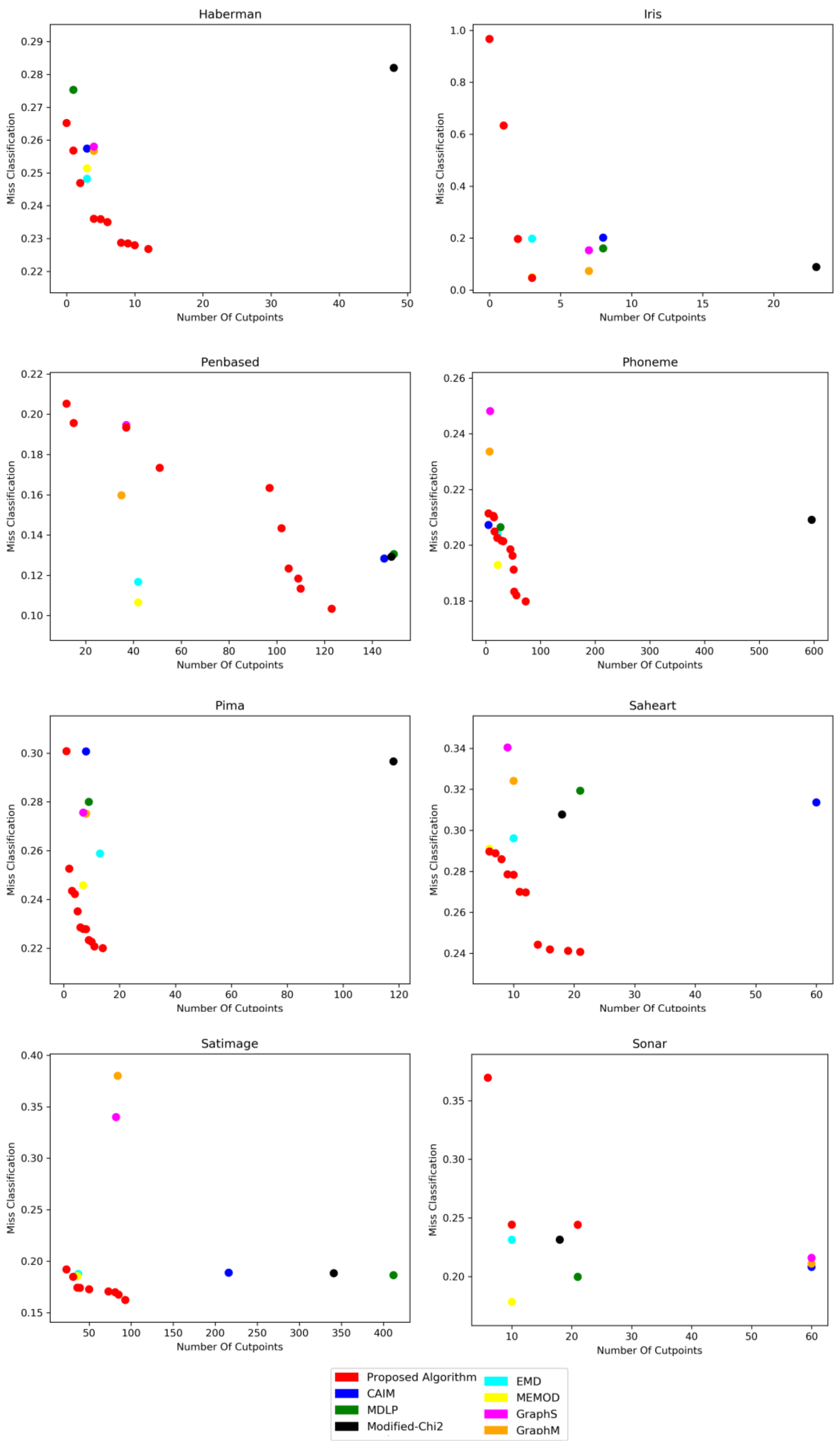


Figure 2. Projection of non-dominated solutions obtained by the proposed algorithm for the considered datasets (Part-2).

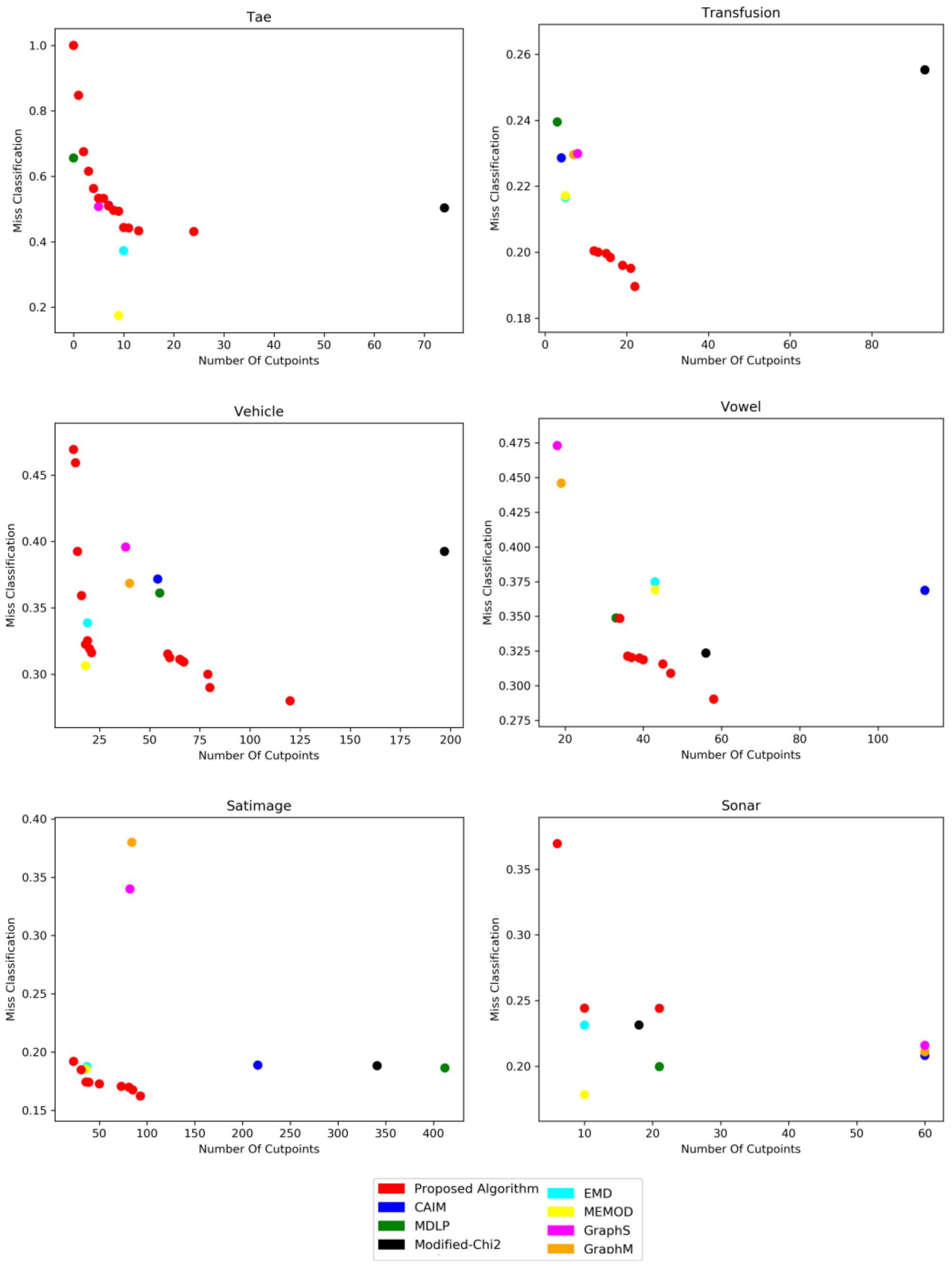


Figure 3. Projection of non-dominated solutions obtained by the proposed algorithm for the considered datasets (Part-3).

Table 4. The number of cut points obtained by algorithms.

	CAIM	MDLP	Modified-Chi2	EMD	MEMOD	GraphM	GraphS	Proposed algorithm
abalone	216	41	3265	102	22	17	**16**	17
appendicits	7	6	40	3	**2**	7	18	4
balance	8	**4**	12	9	9	8	8	9
bupa	6	**1**	130	11	11	6	7	9
contraceptive	15	10	44	26	8	9	9	**6**
glass	13	**12**	53	54	15	15	14	14
haberman	3	**1**	48	3	3	4	4	10
iris	8	8	23	**3**	**3**	7	7	**3**
penbased	145	149	148	42	42	**35**	37	37
phoneme	**5**	27	596	22	22	7	8	**5**
pima	8	9	118	13	**7**	8	**7**	11
saheart	60	21	18	10	**6**	10	9	12
satimage	216	412	341	37	**36**	84	82	**36**
sonar	60	21	18	**10**	**10**	60	60	**10**
tae	7	**0**	74	10	9	5	5	10
transfusion	4	**3**	93	5	5	7	8	12
vehicle	54	55	197	19	**18**	40	38	20
vowel	112	33	56	43	43	19	**18**	58
wine	26	24	13	**3**	**3**	12	13	12
yeast	57	13	179	33	24	**8**	**8**	17
mean	51.5	42.5	273.3	22.9	**14.9**	18.4	18.8	15.05

Table 5. The Classification accuracy obtained by NB.

	CAIM		**MDLP**		**Modified-Chi2**		**EMD**		**MEMOD**		**GraphM**		**GraphS**		**Proposed method**	
	Acc	std	Acc	std	Acc	std	Acc	std	Acc	Std	Acc	std	Acc	std	Acc	std
abalone	0.2511	0.0189	0.233	0.0164	0.1928	0.0196	0.2436	0.0195	**0.2654**	0.0199	0.2176	0.0202	0.2123	0.0296	0.2176	0.0126
appendicitis	0.8023	0.0194	0.8023	0.0194	0.7633	0.0579	0.8023	0.0194	0.8023	0.0194	0.8023	0.0091	0.8023	0.0091	**0.8117**	0.0348
balance	0.8017	0.0379	0.7232	0.0419	**0.8315**	0.0274	0.5762	0.0597	0.8248	0.0388	0.6781	0.0664	0.5273	0.0498	0.8288	0.0323
bupa	0.6186	0.029	0.5798	0.0084	0.6364	0.0719	0.6346	0.0466	0.7011	0.056	0.6764	0.0748	0.6626	0.0756	**0.7111**	0.0125
contraceptive	0.4661	0.0366	0.4833	0.0362	0.5075	0.0426	0.4812	0.0373	**0.5268**	0.0332	0.4966	0.0345	0.4984	0.0314	**0.5268**	0.0309
glass	0.6745	0.079	0.6319	0.0815	0.6663	0.0945	0.6333	0.0929	0.6564	0.1016	0.6617	0.0817	0.6599	0.0861	**0.6754**	0.0953
haberman	0.7359	0.0262	0.7353	0.0069	0.7379	0.0702	0.7223	0.023	0.7353	0.0069	0.7272	0.0344	0.7374	0.0447	**0.7415**	0.0537
iris	0.6433	0.0911	0.7347	0.1115	0.88	0.0736	0.7933	0.0808	**0.9533**	0.0512	0.91	0.0669	0.748	0.1044	**0.9533**	0.0494
penbased	0.7838	0.0113	0.7757	0.0096	0.7805	0.0095	0.8082	0.011	**0.8401**	0.0103	0.7341	0.0114	0.6647	0.0108	0.6647	0.0113
phoneme	**0.7885**	0.0159	0.7611	0.0162	0.7118	0.0185	0.7355	0.0173	0.7754	0.0172	0.7379	0.0153	0.7065	0.0005	0.7883	0.0456
pima	0.6635	0.0266	0.6663	0.029	0.6776	0.0523	0.6988	0.0319	0.7511	0.0367	0.7069	0.0436	0.7133	0.036	**0.7567**	0.0326
saheart	0.669	0.0393	0.6537	0.003	0.7067	0.0601	0.6775	0.035	0.6993	0.0483	0.6553	0.0124	0.6549	0.0054	**0.7117**	0.0489
satimage	0.7729	0.0108	0.7711	0.0122	0.7686	0.0119	0.7688	0.0139	**0.8053**	0.0117	0.5416	0.017	0.6215	0.0147	0.7953	0.0182
sonar	0.7824	0.0895	**0.8297**	0.0697	0.7545	0.082	0.7038	0.0818	0.7932	0.0778	0.8009	0.088	0.7822	0.0949	0.7207	0.0806
tae	0.4262	0.1361	0.344	0.017	0.4202	0.0102	0.5415	0.042	**0.8082**	0.017	0.4496	0.0128	0.4647	0.117	0.4402	0.1038
transfusion	0.7687	0.0182	0.7621	0.0041	0.7337	0.0424	0.7636	0.0082	0.7621	0.0041	0.7631	0.047	0.7631	0.0047	**0.7847**	0.0125
vehicle	0.5774	0.0502	0.5701	0.0526	0.5119	0.0489	0.6055	0.0441	0.6681	0.0422	0.5914	0.0466	0.5583	0.0438	**0.6692**	0.0362
vowel	0.4594	0.0495	0.5438	0.0504	0.5498	0.0483	0.4898	0.0537	0.551	0.0396	0.4255	0.0493	0.3689	0.0413	**0.6161**	0.0393
wine	0.9214	0.0576	0.9237	0.0173	0.9276	0.0564	0.9045	0.0554	0.9138	0.0629	0.4681	0.0331	0.4901	0.0314	**0.9471**	0.0513
yeast	0.5428	0.0378	0.491	0.0403	0.5315	0.0383	0.5028	0.0399	**0.5809**	0.0323	0.4531	0.0393	0.4662	0.672	0.4378	0.0362
mean	0.6575	0.0440	0.6508	0.0322	0.6645	0.0468	0.6544	0.0407	**0.7207**	0.0364	0.6249	0.0402	0.6051	0.0752	0.6894	0.0419

Table 6. The classification accuracy obtained by C4.5.

	CAIM		**MDLP**		**Modified-Chi2**		**EMD**		**MEMOD**		**GraphM**		**GraphS**		**Proposed method**	
	Acc	std	Acc	std	Acc	std	Acc	Std	Acc	Std	Acc	std	Acc	std	Acc	std
abalone	0.2333	0.0209	0.239	0.0190	0.2039	0.0200	0.2294	0.0191	**0.2627**	0.0203	0.2260	0.0203	0.2082	0.0307	0.2450	0.0295
appendicitis	0.8669	0.0720	0.8738	0.0715	0.872	0.0863	0.8838	0.0890	0.8961	0.0812	0.8687	0.0896	0.8832	0.0823	**0.9153**	0.0818
balance	0.7519	0.0412	0.7003	0.0347	0.7969	0.0482	0.6017	0.0507	0.8001	0.0447	0.7945	0.0365	0.7537	0.0467	**0.8415**	0.0327
bupa	0.7036	0.0590	0.6319	0.0734	0.6335	0.0837	0.7113	0.0861	0.6862	0.0779	0.6216	0.0752	0.6084	0.0740	**0.7069**	0.0369
contraceptive	0.4869	0.0370	0.5488	0.0371	0.4978	0.0382	0.5292	0.0404	**0.5580**	0.0414	0.4911	0.0344	0.5076	0.0365	0.5361	0.0311
glass	0.7009	0.0922	**0.7616**	0.0730	0.7457	0.0936	0.7415	0.0870	0.7126	0.0955	0.6793	0.0950	0.6829	0.0943	0.7534	0.0859
haberman	0.7494	0.0495	0.7141	0.0476	0.6982	0.0603	0.7813	0.0509	0.7620	0.0497	0.7594	0.0424	0.7466	0.0498	**0.7866**	0.0548
iris	0.9533	0.0487	0.9447	0.0604	0.9427	0.055	0.8113	0.0929	0.9467	0.0573	0.9433	0.0493	0.9460	0.0469	**0.9533**	0.0521
penbased	0.9597	0.0053	**0.9634**	0.0056	0.9611	0.0065	0.9584	0.0059	0.9469	0.0063	0.9465	0.0070	0.9462	0.0070	0.9486	0.037
phoneme	0.7970	0.0171	0.826	0.0157	**0.8700**	0.013	0.8568	0.0122	0.8288	0.0149	0.7949	0.0170	0.7972	0.0152	0.7890	0.0587
pima	0.7351	0.0394	0.7737	0.0442	0.7292	0.0463	0.7836	0.0432	0.7573	0.0478	0.7428	0.0429	0.7356	0.0400	**0.7868**	0.0384
saheart	0.7039	0.0537	0.7078	0.0508	0.6779	0.0533	0.7303	0.0597	0.7188	0.0576	0.6965	0.051	0.6643	0.0576	**0.7490**	0.0485
satimage	0.8496	0.0126	0.8561	0.0128	0.8548	0.0125	**0.8561**	0.0127	0.8238	0.0127	0.6985	0.0158	0.6987	0.0152	0.8561	0.0197
sonar	0.8013	0.0876	0.7710	0.0737	0.7828	0.0832	0.8336	0.0776	**0.8500**	0.0771	0.7772	0.0841	0.7860	0.0920	0.7909	0.0887
tae	0.5512	0.1065	0.3440	0.0170	0.5728	0.1147	0.7133	0.0450	**0.8430**	0.0700	0.4855	0.1421	0.5208	0.1236	0.6791	0.1108
transfusion	0.7741	0.0161	0.7590	0.0173	0.7557	0.0388	0.8032	0.0395	0.8038	0.0307	0.7778	0.0222	0.7771	0.0205	**0.8145**	0.0148
vehicle	0.6793	0.040	0.7076	0.0432	0.7031	0.0406	0.7172	0.037	**0.7191**	0.0430	0.6717	0.0476	0.6502	0.0418	0.6923	0.0474
vowel	**0.8033**	0.0411	0.7586	0.0410	**0.8033**	0.0367	0.7607	0.0443	0.7110	0.0421	0.6826	0.0426	0.6849	0.038	**0.8033**	0.0334
wine	0.9144	0.0684	0.9416	0.0644	0.9599	0.0535	**0.9621**	0.0474	0.9195	0.0587	0.4295	0.0394	0.452	0.0384	0.9748	0.0649
yeast	0.5507	0.0378	**0.5961**	0.0382	0.5089	0.0415	0.5133	0.0383	0.5538	0.0361	0.5454	0.0423	0.5494	0.0635	0.5229	0.0394
mean	0.7283	0.0473	0.7210	0.0420	0.7285	0.0513	0.7389	0.0489	0.7550	0.0483	0.6816	0.0498	0.6800	0.0507	**0.7573**	0.0505

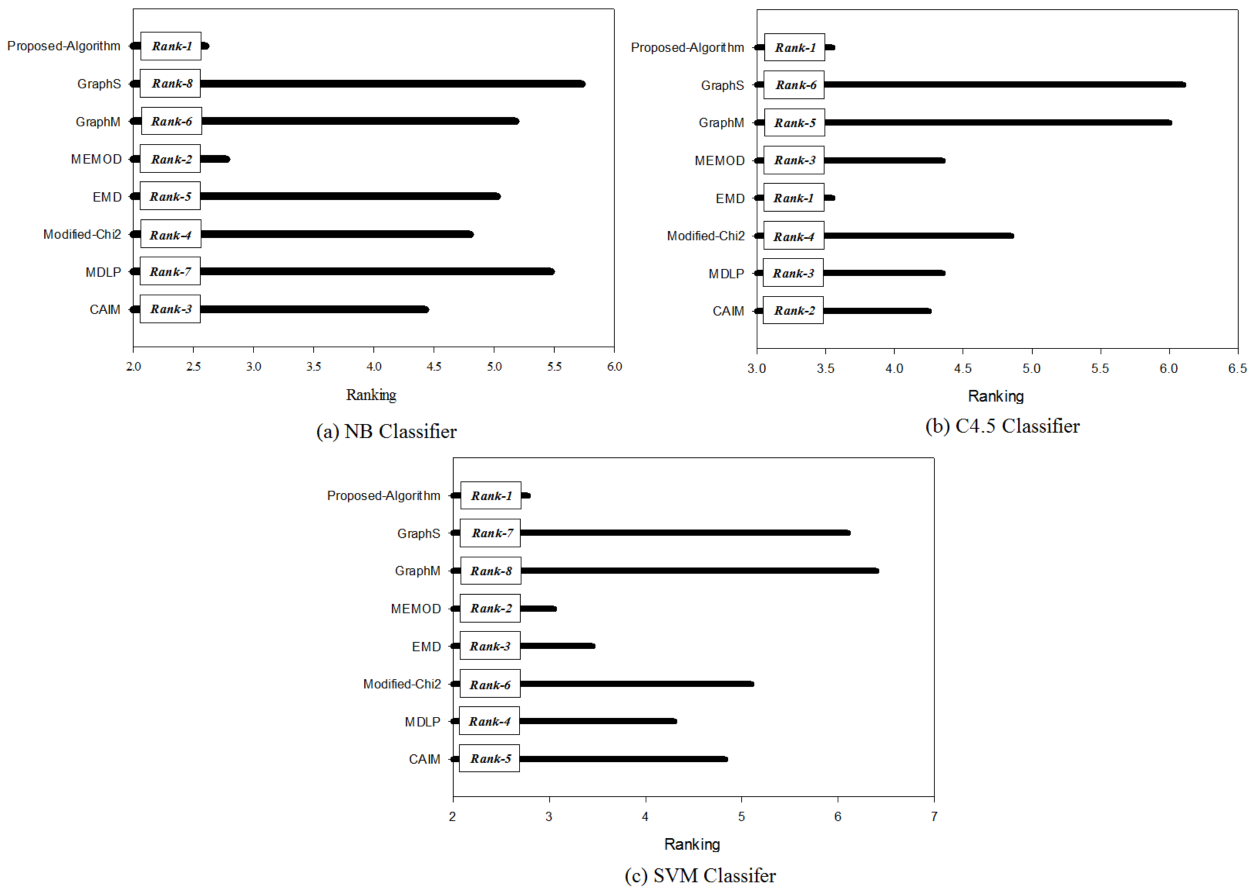
Table 7. The classification accuracy obtained by SVM.

	CAIM		**MDLP**		**Modified-Chi2**		**EMD**		**MEMOD**		**GraphM**		**GraphS**		**Proposed method**	
	Acc	std	Acc	std	Acc	std	Acc	Std	Acc	Std	Acc	std	Acc	std	Acc	std
abalone	0.2333	0.0209	0.239	0.0190	0.2039	0.0200	0.2294	0.0191	**0.2627**	0.0203	0.2260	0.0203	0.2082	0.0307	0.2450	0.0295
appendicitis	0.8669	0.0720	0.8738	0.0715	0.872	0.0863	0.8838	0.0890	0.8961	0.0812	0.8687	0.0896	0.8832	0.0823	**0.9153**	0.0818
balance	0.7519	0.0412	0.7003	0.0347	0.7969	0.0482	0.6017	0.0507	0.8001	0.0447	0.7945	0.0365	0.7537	0.0467	**0.8415**	0.0327
bupa	0.7036	0.0590	0.6319	0.0734	0.6335	0.0837	0.7113	0.0861	0.6862	0.0779	0.6216	0.0752	0.6084	0.0740	**0.7069**	0.0369
contraceptive	0.4869	0.0370	0.5488	0.0371	0.4978	0.0382	0.5292	0.0404	**0.5580**	0.0414	0.4911	0.0344	0.5076	0.0365	0.5361	0.0311
glass	0.7009	0.0922	**0.7616**	0.0730	0.7457	0.0936	0.7415	0.0870	0.7126	0.0955	0.6793	0.0950	0.6829	0.0943	0.7534	0.0859
haberman	0.7494	0.0495	0.7141	0.0476	0.6982	0.0603	0.7813	0.0509	0.7620	0.0497	0.7594	0.0424	0.7466	0.0498	**0.7866**	0.0548
iris	0.9533	0.0487	0.9447	0.0604	0.9427	0.055	0.8113	0.0929	0.9467	0.0573	0.9433	0.0493	0.9460	0.0469	**0.9533**	0.0521
penbased	0.9597	0.0053	**0.9634**	0.0056	0.9611	0.0065	0.9584	0.0059	0.9469	0.0063	0.9465	0.0070	0.9462	0.0070	0.9486	0.037
phoneme	0.7970	0.0171	0.826	0.0157	**0.8700**	0.013	0.8568	0.0122	0.8288	0.0149	0.7949	0.0170	0.7972	0.0152	0.7890	0.0587
pima	0.7351	0.0394	0.7737	0.0442	0.7292	0.0463	0.7836	0.0432	0.7573	0.0478	0.7428	0.0429	0.7356	0.0400	**0.7868**	0.0384
saheart	0.7039	0.0537	0.7078	0.0508	0.6779	0.0533	0.7303	0.0597	0.7188	0.0576	0.6965	0.051	0.6643	0.0576	**0.7490**	0.0485
satimage	0.8496	0.0126	0.8561	0.0128	0.8548	0.0125	**0.8561**	0.0127	0.8238	0.0127	0.6985	0.0158	0.6987	0.0152	0.8561	0.0197
sonar	0.8013	0.0876	0.7710	0.0737	0.7828	0.0832	0.8336	0.0776	**0.8500**	0.0771	0.7772	0.0841	0.7860	0.0920	0.7909	0.0887
tae	0.5512	0.1065	0.3440	0.0170	0.5728	0.1147	0.7133	0.0450	**0.8430**	0.0700	0.4855	0.1421	0.5208	0.1236	0.6791	0.1108
transfusion	0.7741	0.0161	0.7590	0.0173	0.7557	0.0388	0.8032	0.0395	0.8038	0.0307	0.7778	0.0222	0.7771	0.0205	**0.8145**	0.0148
vehicle	0.6793	0.040	0.7076	0.0432	0.7031	0.0406	0.7172	0.037	**0.7191**	0.0430	0.6717	0.0476	0.6502	0.0418	0.6923	0.0474
vowel	**0.8033**	0.0411	0.7586	0.0410	**0.8033**	0.0367	0.7607	0.0443	0.7110	0.0421	0.6826	0.0426	0.6849	0.038	**0.8033**	0.0334
wine	0.9144	0.0684	0.9416	0.0644	0.9599	0.0535	**0.9621**	0.0474	0.9195	0.0587	0.4295	0.0394	0.452	0.0384	0.9748	0.0649
yeast	0.5507	0.0378	**0.5961**	0.0382	0.5089	0.0415	0.5133	0.0383	0.5538	0.0361	0.5454	0.0423	0.5494	0.0635	0.5229	0.0394
mean	0.7283	0.0473	0.7210	0.0420	0.7285	0.0513	0.7389	0.0489	0.7550	0.0483	0.6816	0.0498	0.6800	0.0507	**0.7573**	0.0505

3.2. Non-parametric statistical tests

The non-parametric statistical tests were used to confirm the statistical significance and to analyze the difference between the discretizers. For this purpose, a two-step procedure was used. In the first step, a non-parametric statistical test such as the Friedman's test was performed to rank the algorithms based on their performance [23]. According to the null hypothesis in this test, the algorithms are equivalent. Thus by rejecting the null hypothesis, one can conclude the statistically significant differences. In the second step, in order to examine the existence of a pairwise difference, an algorithm with the best rank was selected as the control algorithm for comparison with other algorithms, and then the post-hoc non-parametric statistical tests were performed to compare the statistical difference between the control algorithm (the best algorithm) with the other algorithms [24]. The post-hoc statistical tests used in this work included Holm [25], Hochberg [26], Hommel [27], Holland [28], Rom [29], and Finner [30].

In this work, there were eight discretizers for comparing. The experiments were designed in such a way that the statistical significance of the accuracy of discretizers and the number of cut points obtained from them were tested. In order to achieve this goal, the tests were performed on the classification accuracy and the number of cut points obtained from the discretizer on 20 datasets. In figure 4, the results of the Friedman test are shown. This test was applied with a level of confidence $\alpha = 0.05$. According to the results shown in the figures, the proposed algorithm ranked first in all the three classifiers, and therefore, it was chosen as the control algorithm.



(a) NB Classifier

(b) C4.5 Classifier

(c) SVM Classifer

Figure 4. Average ranks obtained by each algorithm in the Friedman test.

Tables 8, 9, and 10 show the values obtained from applying the post-hoc tests on the results of the Friedman test. The tables show that there is a statistically significant difference between the proposed algorithm and the other algorithms. Therefore, the proposed algorithm has a higher accuracy than the other algorithms, and can be considered as an appropriate algorithm for discretization.

Table 8. Post Hoc comparison Table for $\alpha = 0.05$ (Friedman for NB classfier).

Algorithm	Holm/ Hochberg/ Hommel	Holland	Rom	Finner
GraphS	0.007143	0.007301	0.007513	0.007301
MDLP	0.008333	0.008512	0.008764	0.014548
GraphM	0.010000	0.010206	0.010515	0.021743
EMD	0.012500	0.012741	0.013109	0.028885
Modified-Chi2	0.016667	0.016952	0.016667	0.035975
CAIM	0.025000	0.025321	0.025000	0.043013
MEMOD	0.050000	0.050000	0.050000	0.050000

Table 9. Post Hoc comparison Table for $\alpha = 0.05$ (Friedman for C4.5 classfier).

Algorithm	Holm/ Hochberg/ Hommel	Holland	Rom	Finner
GraphM	0.007143	0.007301	0.007513	0.007301
GraphS	0.008333	0.008512	0.008764	0.014548
Modified-Chi2	0.010000	0.010206	0.010515	0.021743
CAIM	0.012500	0.012741	0.013109	0.028885
MDLP	0.016667	0.016952	0.016667	0.035975
EMD	0.025000	0.025321	0.02500	0.043013
MEMOD	0.050000	0.050000	0.050000	0.050000

Table 10. Post Hoc comparison Table for $\alpha = 0.05$ (Friedman for SVM classfier).

Algorithm	Holm/ Hochberg/ Hommel	Holland	Rom	Finner
GraphS	0.007143	0.007301	0.007513	0.007301
GraphM	0.008333	0.008512	0.008764	0.014548
MDLP	0.010000	0.010206	0.010515	0.021743
MEMOD	0.012500	0.012741	0.013109	0.028885
CAIM	0.016667	0.016952	0.016667	0.035975
Modified-Chi2	0.025000	0.025321	0.02500	0.043013
MEMOD	0.050000	0.050000	0.050000	0.050000

4. Conclusions and future works

In this paper, a new evolutionary multi-objective algorithm has been proposed using the NSGA-III algorithm for multivariate discretization. This algorithm utilizes the normalized cut criterion. The proposed algorithm optimizes three objective functions simultaneously. This algorithm not only reduces the number of selected cut points and classification errors but also the relationships between the features' values and the nature of the data is maintained. The performance of the proposed algorithm was compared with other algorithms in the literature. The results obtained indicate that the proposed algorithm is better than the other algorithms in the classification accuracy. On the other hand, since the proposed algorithm is trying to maintain the nature of the data, it obtains the second rank in terms of the number of selected cut points. Our algorithm is capable of solving several solutions (the Pareto front), which allows the user to choose solutions to their problem from solutions in the Pareto front.

The two-step non-parametric statistical tests were used to show better results. Based on the non-parametric statistical tests, the proposed algorithm ranked first among the other algorithms. Achieving the first rank means that the proposed algorithm performs better than the other algorithms. Then the algorithms were compared in pairwise, which showed that the proposed algorithm was significantly better than the other algorithms.

The future research works can provide unsupervised discretization based on evolutionary multi-objective algorithms, taking into account the objectives of the number of selected cut points, inter cluster and intra cluster. Also EAs with high-dimensional data may require a lot of running time. Using the parallelization techniques can greatly improve the runtime.

References

[1] Pouramini A, Khaje Hassani S, Nasiri S (2018). Data Extraction using Content-Based Handles, Journal of AI and Data Mining, vol. 6, pp. 399-407. doi:10.22044/jadm.2017.990

[2] Ramírez-Gallego S, García S, Benítez JM, Herrera F (2018). A distributed evolutionary multivariate discretizer for big data processing on apache spark, Swarm and Evolutionary Computation, vol. 38, pp. 240-250.

[3] García-Gil D, Ramírez-Gallego S, García S, Herrera F (2018). Principal components analysis random discretization ensemble for big data, Knowledge-Based Systems, vol. 150, pp. 166-174.

[4] Ramírez-Gallego S, García S, Herrera F (2018). Online entropy-based discretization for data streaming classification, Future Generation Computer Systems, vol. 86, pp. 59-70.

[5] Ramírez-Gallego S, García S, Benítez JM, Herrera F (2016). Multivariate discretization based on evolutionary cut points selection for classification, IEEE transactions on cybernetics, vol. 46, pp. 595-608.

[6] Tahan MH, Asadi S (2018). MEMOD: a novel multivariate evolutionary multi-objective discretization, Soft Computing, vol. 22, pp. 301-323. doi:10.1007/s00500-016-2475-5

[7] Tahan MH, Asadi S (2018). EMDID: Evolutionary multi-objective discretization for imbalanced datasets, Information Sciences, vol. 432, pp. 442-461. doi: 10.1016/j.ins.2017.12.023

[8] Fayyad U, Irani K (1993). Multi-interval discretization of continuous-valued attributes for classification learning.

[9] Kurgan LA, Cios KJ (2004). CAIM discretization algorithm, IEEE transactions on Knowledge and Data Engineering, vol. 16, pp. 145-153.

[10] Tay FE, Shen L (2002). A modified chi2 algorithm for discretization, IEEE Transactions on Knowledge & Data Engineering, pp. 666-670.

[11] Sriwanna K, Boongoen T, Iam-On N (2018). Graph clustering-based discretization approach to microarray data, Knowledge and Information Systems, pp. 1-28.

[12] Kim K-j, Han I (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index, Expert systems with Applications, vol. 19, pp. 125-132.

[13] Yang H, Wang J, Shao X, Wang NS (2007). Information System Continuous Attribute Discretization Based on Binary Particle Swarm Optimization. In: Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007), 2007. IEEE, pp. 173-177.

[14] García S, López V, Luengo J, Carmona CJ, Herrera F (2012). A Preliminary Study on Selecting the Optimal Cut Points in Discretization by Evolutionary Algorithms. In: ICPRAM (1), 2012. pp. 211-216.

[15] Zamudio-Reyes R, Cruz-Ramírez N, Mezura-Montes E (2017). A Multivariate Discretization Algorithm Based on Multiobjective Optimization. In: 2017 International Conference on Computational Science and Computational Intelligence (CSCI), 2017. IEEE, pp. 375-380.

[16] Lotfi S, Karimi F (2017). A Hybrid MOEA/D-TS for Solving Multi-Objective Problems, Journal of AI and Data Mining, vol. 5, pp. 183-195. doi:10.22044/jadm.2017.886

[17] Coello CAC, Lamont GB, Van Veldhuizen DA (2007). Evolutionary algorithms for solving multi-objective problems, vol 5. Springer. doi:10.1007/978-0-387-36797-2.

[18] Ngatchou P, Zarei A, El-Sharkawi A (2005). Pareto multi objective optimization. In: Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems, 2005. IEEE, pp. 84-91. doi:10.1109/ISAP.2005.1599245.

[19] Tahan MH, Ghasemzadeh M (2019). An evolutionary multi-objective algorithm for constructing balanced spanning tree, Journal of Soft Computing and Information Technology (JSCIT).

[20] Sriwanna K, Boongoen T, Iam-On N (2017). Graph clustering-based discretization of splitting and merging methods(GraphS and GraphM),Human-centric Computing and Information Sciences, vol. 7, p. 21.

[21] Deb K, Jain H (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints, IEEE Transactions on Evolutionary Computation, vol. 18, pp. 577-601.

[22] Blake CL (1998). UCI Repository of machine learning databases, Irvine, University of California, http://www ics uci edu/~ mlearn/MLRepository html.

[23] Sheskin DJ (2003). Handbook of parametric and nonparametric statistical procedures. crc Press,

[24] García S, Fernández A, Luengo J, Herrera F (2009). A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability, Soft Computing, vol. 13, p. 959.

[25] Holm S (1979). A simple sequentially rejective multiple test procedure, Scandinavian journal of statistics, pp. 65-70.

[26] Hochberg Y (1988). A sharper Bonferroni procedure for multiple tests of significance, Biometrika, vol. 75, pp. 800-802.

[27] Hommel G (1988). A stagewise rejective multiple test procedure based on a modified Bonferroni test, Biometrika, vol. 75, pp. 383-386.

[28] Holland BS, Copenhaver MD (1987). An improved sequentially rejective Bonferroni test procedure, Biometrics, pp. 417-423.

[29] Rom DM (1990). A sequentially rejective test procedure based on a modified Bonferroni inequality, Biometrika, vol. 77, pp. 663-665.

[30] Finner H (1993). On a monotonicity problem in step-down multiple test procedures, Journal of the American Statistical Association, vol. 88, pp. 920-923.

نشریه هوش مصنوعی و داده کاوی

الگوریتم گسسته سازی چند هدفه تکاملی مبتنی بر برش نرمال

مرضیه حاجی زاده طحان و محمد قاسم زاده*

[۱]گروه مهندسی کامپیوتر، دانشگاه یزد، یزد، یزد، ایران.

ارسال ۲۰۱۹/۰۶/۰۱؛ بازنگری ۲۰۱۹/۱۰/۲۰؛ پذیرش ۲۰۱۹/۱۱/۲۹

چکیده:

مدل‌های یادگیری و نتایج مربوط به آن به کیفیت داده ورودی وابسته هستند. اگر داده‌های خام به درستی پاک‌سازی و ساخت‌یافته نشوند نتایج اشتباهی را به همراه خواهند داشت. از این رو گسسته‌سازی به عنوان یکی از تکنیک‌های پیش‌پردازش نقش مهمی را در فرآیندهای یادگیری ایفا می‌کند. مهم‌ترین چالش در فرآیند گسسته‌سازی این است که با کاهش تعداد مقادیر ویژگی‌ها، ارتباطات بین ویژگی‌ها حفظ شود و دقت الگوریتم طبقه‌بندی افزایش یابد. در این مقاله یک الگوریتم چندهدفه تکاملی جدید ارائه شده است. الگوریتم پیشنهادی از سه تابع هدف برای دستیابی به گسسته‌سازی با کیفیت بالا استفاده می‌کند. هدف اول و دوم به ترتیب تعداد نقاط برش انتخابی و خطای کلاس‌بندی را کاهش می‌دهد. هدف سوم یک معیار جدید به نام برش نرمال معرفی می‌کند که با استفاده از آن ارتباطات بین ویژگی‌ها و مقادیر آن‌ها و در نتیجه طبیعت داده‌ها حفظ می‌گردد. عملکرد الگوریتم پیشنهادی با استفاده از بیست مجموعه‌داده الگو مورد آزمایش قرار گرفت. بنابر مقایسه‌ها و نتایج آزمون‌های آماری ناپارامتری، الگوریتم پیشنهادی عملکرد بهتری نسبت به سایر روش‌های موجود در ادبیات دارد. بنابراین نتایج نویدبخش این است که الگوریتم پیشنهادی الگوریتم گسسته‌سازی مناسبی در مسائل طبقه‌بندی است.

کلمات کلیدی: گسسته سازی، چندهدفه، تکاملی، برش نرمال، چندمتغیره.