Cluster quality assessment by the modified Renyi-Clip*X* algorithm

Dalia Baziukė, Aleksas Narščius

***Abstract* —** This paper presents the modified Renyi-CLIPx clustering algorithm and shows that with this algorithm more accurate groupings can be obtained, it gives higher accuracy rates as well. The combination of Renyi entropy based clustering and CLIP3, CLIP4 rule derivation algorithms is used to discover clusters and create rules to explain them. The algorithm itself contains refinements that are used while improving the goodness of obtained clusters and rules. Results on researching the influence of data discretization and so called negative examples data set to the rule complexity and cluster quality are presented as well.

***Key words*—** Renyi entropy, CLIP3, CLIP4, clustering, clustering rules, algorithm

I. Introduction

In machine learning the notion of conceptual clustering is used to distinguish it from typical clustering. The general procedure of conceptual clustering consists of two basic steps. The first is clustering, which finds clusters in a given data set, and the second is a characterization, which generates a concept description for each cluster found by clustering and is expressed by logical rules [3], [4], [6].

We were investigating several clustering and rule derivation algorithms to be able accomplish some specific tasks concerning state recognition problems that are described in [1]. The result we came out with was the combination of known algorithms with some additional refinements that led to more accurate results.

Here, in the introductory part, the main algorithms are compared according to four aspects [2]:

- cluster initiation,
- similarity (difference) between object evaluation,
- optimal number of cluster selection,
- complexity of algorithm.

Cluster initiation step is often important for the most clustering algorithms. This is because stability and goodness of final cluster structure strongly depends on prototypes of initial clusters. How stable the computed clusters will be will depend on the method used to define similarity and dissimilarity between objects in the data set.

Defining optimal number of clusters is often discussed problem referred in many papers [7]. For the automatic cluster formation purposes it is usually desired that algorithm can decide on it. Table 1 gives comparison results of Renyi entropy, CLIP3, CLIP4 and ITERATES clustering algorithms according these aspects.

TABLE I
Comparison of the algorithms

	Renyi entropy	ITERATE	CLIP3, CLIP4
Cluster initiation	Centers are randomly selected. Point nearest to any initiated cluster is included into it.	Is based on the ADO sorting divisive hierarchical clustering.	Initial data set is divided according the decision tree construction principals.
Determining similarities (dissimilarities) between objects	Is based on the entropy computation.	Is based on the category utility measure.	Is based on the solution of an integer programming (*IP*) task.
Setting the optimal cluster number	Is based on the in-between cluster entropy change monitoring.	Is based on the category utility measure change monitoring.	Is based on the noise and best rule stopping thresholds.
Algorithm Complexity (*N* – number of objects)	$O(N^2)$	$O(N^3)$	$O(N^2)$

If comparing Renyi entropy clusterization with ITERATE, the first algorithm is easier to implement. The latter is harder in implementation, but it generates qualitatively better and stable clusters [2]. CLIP algorithms are two class rule learning algorithms and will be applied in the proposed algorithm. CLIP3 and CLIP4 conceptual clustering algorithms generate rules that are being used to assign objects into clusters [4], [5].

II. Combining Renyi Entropy Clustering and CLIPx Rules Derivation Algorithm

Combining clustering and rule derivation algorithms and, adding some refinements to improve cluster goodness, and removing overlapping rules we invent a modified algorithm (Fig. 1). It is named modified Renyi-CLIPx based on the two known algorithms [4], [5], [6] that are in the core and is modified, because we put some refinements according object redistribution between clusters and final rule definition.

According the flow depicted in the Fig. 1, Renyi entropy clustering algorithm divides elements from the initial data set into different clusters, and CLIP3 or CLIP4 algorithm generates clustering rules. CLIP3 and CLIP4 require positive (POS) and negative (NEG) data subsets to be loaded. For that, we keep the following setting:

1. Data subset POS contains elements from the cluster to which the describing rules are being derived.
2. Elements from the rest clusters are included into the NEG data subset.

During the rules generation process, every cluster gets a unique collection of rules indicating feature values that are not allowed for an object to have in particular cluster. POS and NEG data subsets are being composed for every cluster to be learned.

The algorithm relationship module implements communication between Renyi entropy and CLIP3/CLIP4 parts. It ensures transformation of results and rendition through interconnected algorithms. Modified Renyi-CLIPx conceptual clustering algorithm is capable to define true number of clusters, construct these clusters, and generate clustering rules of reasonable quality. The algorithm implementation has four main parts including modules responsible for:

- data preprocessing,
- redistribution of objects in-between clusters,
- graphical presentation of clusters,
- validation of clustering rules.

III. Data Preprocessing

Module of data preparation is designed to enhance performance of clustering algorithms, ensure preconditions to construct qualitative clusters and less overlapping rules. During the data preparation procedure it is suggested to perform such operations:

- detect and remove statistical exceptions,
- remove repetitive objects,
- discretize data,
- normalize data.

The operation can be chosen depending on a given data set, requirements and clustering algorithm. After the normalization or discretization of a given data set it is necessary to examine data for repetitive objects. For example, after data normalization or discretization it might happen that objects with the same feature values are included into different clusters. In such a case it will be impossible to generate rules for existing concepts. The way how to overcome this issue would be removing repetitive objects or choosing wider discretization interval.

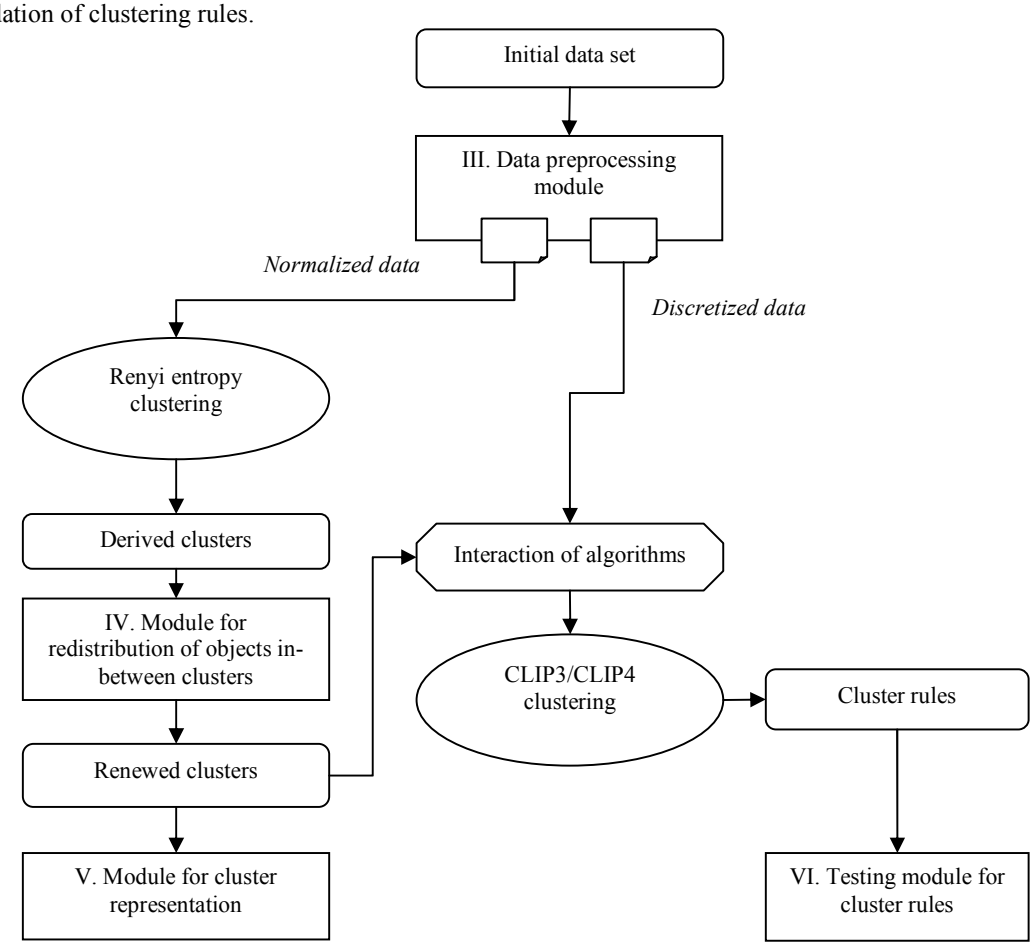

Fig 1. Data processing flow in the modified Renyi-CLIPx clustering algorithm [9]

IV. Redistributing Objects In-between Clusters

Module for redistributing objects in-between clusters improves the quality of inside cluster structure. Several methods can be used to examine the quality of obtained clusters. The choice of corresponding method depends on characteristics of a given data set. For example, *silhouette* measures [8] and graphical it representation can be used. Graphs of silhouette measurement presents how near to different clusters objects' current object is [3]. In our case distance from one object to another was measured using Euclidian distance. In ideal case (when clusters are maximum isolated and all objects are assigned into right clusters) this measure is equal to +1. If it is not clear to which cluster an object should belong this measure will have value equal to 0. When it is clear that object is assigned into wrong cluster, silhouette measure is equal -1. The measure can have any value in the interval [-1; 1].

Fig. 2 is depicting silhouette measure graphs for three clusters. In Fig. 2 b) silhouette measure is positive for all clusters. This means that all objects are probably correctly assigned to the right clusters. More closely this measure value is to 1, it is more likely that object is correctly assigned into a cluster. To evaluate overall goodness of obtained cluster structure the average silhouette measure can be used. The average allows to compare different partitioning and to decide on a real number of clusters. If we have built several partitioning for the same data set, we can decide on the natural number of clusters according to the given data. It will correspond to the partitioning with the biggest silhouette measure average value.

According to the properties of a silhouette measure it can be used in the module for redistributing objects in-between clusters. Redistribution of objects is done following the procedure of checking the silhouette value of particular object and deciding does it require reassignment to another cluster. Reassignment is done only with those objects that have silhouette measure zero or below zero. The object is assigned into that cluster whose silhouette measure value increases mostly by adding this new object. The average silhouette measure value increases after the redistribution (Fig. 2, b).

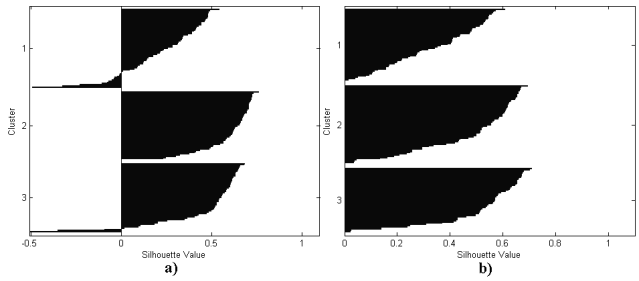

Fig 2. Evaluation of obtained cluster quality:
a) un-redistributed results b) redistributed results

V. Cluster Graphical Representation

Graphical presentation module depicts assignments to the clusters for a given data set. Usually it might be more convenient to analyze clustering results from their graphical representation [3]. In some cases graphical representation might help to determine correctness of the assignment. Good separated groupings have a clear boundary. From the cluster scatter plot one can very clearly define the quality of such separation. If some objects from one cluster get through the boundary and steps to another cluster, we can suppose that clusters are not very well separated.

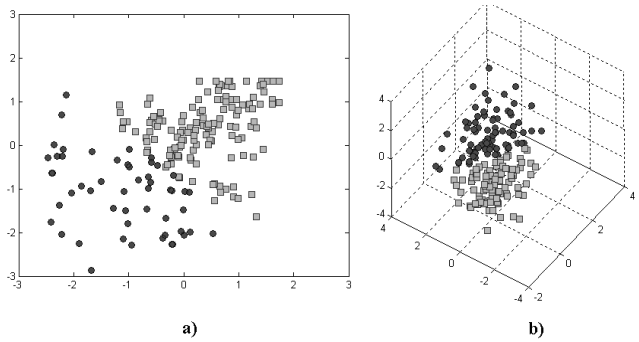

Fig 3. Cases of depicting clusters

Fig. 3 a) and b) shows 2D and 3D cases of cluster representation. In this example normalized students' grades are plotted.

VI. Validation of Clustering Rules

Rules validation module measures the goodness of clustering rules generated by modified conceptual clustering algorithm. These rules have to be of a simple structure as it is possible and take most examples from POS data subset. The rule set is supposed to be good if by applying it all positive examples are assigned only into POS data subset, and negative examples only into NEG data subset. Other parameters to verify goodness of clustering rules can be used as well. For example, the minimum number of different attributes or examples the rules have to take can be set. Also, the minimum complexity (number of logical conditions) in a premise of clustering rule can be defined. Modified Renyi-CLIPx conceptual clustering algorithm ensures goodness of clustering rules by various thresholds (noise threshold, best rule threshold, and stop threshold) and genetic module [5]. At this way we are seeking that rules were not generated only for one example, but for a larger set of examples. At some cases after applying these thresholds clustering rules do not take all POS data subset. If such happens, the expert judgment can be applied. Having ability to select several types of thresholds with modified Renyi-CLIPx conceptual clustering algorithm we are capable to generate well separated clusters, simple to read clustering rules, and clearly returned results.

VII. Deriving Clusters and Clustering Rules

We have used *WINE* data set [7] to check the performance of the modified algorithm in generating clusters and rules. *WINE* set has 178 objects with 13 attributes. It stores information about vintage chemical analyses from Italian regions (13 different chemical elements and 178 observations). All observation values are known. From expert view there are 3 natural clusters. First class contains 59, second – 71, and third – 38 examples from data set. One additional column is for the class which has been assigned by an expert. Original classes are known and will be used to compare grouping results computed by the modified Renyi-CLIPx algorithm.

While investigating the performance of proposed algorithm evaluation was done according to the two criteria. First, do the groupings we get match the original prescriptions to classes? Second, the value of silhouette measure [8]. We were checking performance of the algorithm having in mind two possibilities of selecting: to normalize data or not and redistribute objects or not. The influence of standard deviation and initial number of clusters (K_{init}) to final clustering results was investigated as well. Standard deviation values were taken from interval [0.05; 5.5], and initial number of clusters was set to 20. Experiments were repeated 10 times. Table 2 displays the results.

TABLE II
Results of Evaluation Having Various Values of Standard Deviation and Initial Cluster Numbers

1	2	3	4	5	6
			Evaluation		
Data normalization	Redistribution of results		SD		SD computed, K_{init} preselected
			Preselected	Calculated	
No	No	***VA*:**	41,96%	61,62%	63,51%
		***AG*:**	-0,05	0,67	0,58
		***SI*:**	[0,8; 1,7]	>0	[20; 25]
No	Yes	***VA*:**	62,82%	68,60%	64,66%
		***AG*:**	0,86	0,84	0,86
		***SI*:**	[0,8; 1,7]	>0	[19; 25]
Yes	No	***VA*:**	58,15%	91,08%	77,17%
		***AG*:**	0,08	0,49	0,34
		***SI*:**	[0,9; 1,5]	>0	[24; 30]
Yes	Yes	***VA*:**	89,15%	93,82%	93,82%
		***AG*:**	0,53	0,53	0,55
		***SI*:**	>0	>0	>2

The abbreviations in Table 2 have the following meaning:

- *VA* – averaged match to expert judgment, %
- *AG* –goodness of cluster structure, evaluated by average silhouette measure value.
- *SI* – suggested interval for common aspect. The 4th and 5th column of the table shows intervals for standard deviation. The 6th column displays interval from where initial number of clusters was chosen.
- *SN* – standard deviation.

From the Table 2 it can be inferred that standard deviation and initial number of clusters have different influence for clustering results. Optimal results are reached when initial data set is normalized and objects are being redistributed according the procedure in the object redistribution module. The averaged cluster goodness measure gets lowest values when the normalized data are used. It is supposed that this happens because of the Euclidean distance which is used to evaluated distinctness. With normalized data we will have relatively smaller Euclidian distances between objects. The best results are being reached with normalized data, biggest initial number of clusters, and object redistribution switched-on.

Fig. 4 displays in-between cluster entropy changes of *WINE* data set with initial number of clusters equal to 20, calculated standard deviation from a set of objects, and activated redistribution of resulting clusters.

From the curve in Fig. 4 we get that reducing the number of clusters up to 2 will stimulate a fast increase of the in-between cluster entropy. From there it is decided on the natural number of clusters should be equal to 3. Originally there are three clusters in *WINE* data set. So, the generated number of clusters by the algorithm matches experts' judgment.

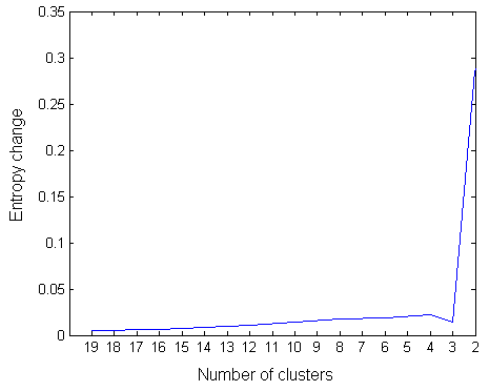

Fig 4. In-between cluster entropy change for *WINE* data

Fig. 2 displays cluster goodness assessment expressed by silhouette measure values for three generated clusters. Not redistributed clusters match to the expert judgment in 87,26%, redistributed ones – 93,82%. This leads to the 60 examples in the first cluster, 65 in the second, and 53 in the third cluster. Such a result is good enough to generate clustering rules. Rules are generated for every cluster and discretized data set, when number of discretized intervals was set to 100.

VIII. Influence of Discretization to the Clustering Rules

Clustering rules can be assessed by several aspects. The complexity of clustering rule can be measured by number of logical conditions. Other important measure of the rule quality is the number of examples the rule is able to cover. Generated clustering rules have to be unique. To ensure this an object is assigned to the POS subset only if it satisfies all generated rules for particular cluster. Cases, when the object from NEG subset can satisfy the rules, are not allowed. For particular experiment with *WINE* data set the modified Renyi-CLIPx clustering algorithm generated rules for every cluster. The premise of each rule examines some particular attributes (features) that can be considered as being main features while deciding about belonging of the object to the cluster. The generated rules have the structure as is given bellow:

- The rule for the first cluster analyses 2 different features, the 1st and 13th column. It has 58 logical conditions in the premise.
- The rule for the second cluster analyses 3 different features, the 4th, 9th, and 13th column. It has 36 logical conditions in the premise.
- The rule for the third cluster analyses 2 different features, the 7th and 10th column. It has 31 logical conditions in the premise.

The generated rules for *WINE* data take all objects from the right POS subset. The number of logical conditions (58 + 36 + 31 = 125) for computer based discrimination is reasonably low. Other positive achievement is that the set of generated rules needs to analyze only 6 different features. The number of logical conjuncts in the rule premise depends on feature values set. The rule becomes more complicated with the increase of the number of conjuncts in the premise. Differences between the same attribute values can be very minimal. For example, values 5.64, 5.65, and 5.68 may be treated as distinct ones. Discretization procedure helps to overcome this problem and reduce the number of logical conjuncts in the rule premise. For example, for the same *WINE* data, but without discretization, the algorithm will generate more complex rules having in total 484 (166 + 143 + 175) logical conditions.

Cases with application of following bounds (thresholds) were investigated. The following thresholds were defined:

- Noise threshold equal to 60%
- Partitioning threshold equal to 2
- Stop threshold equal to 2%
- Minimal number of logical conditions in a rule equal to 3
- Minimal number of distinct features included in a rule premise equal to 3
- Minimal number of examples described by a rule equal to 4

With those bounds we intend to restrict generation of random rules for a small number of examples.

Fig 5 shows general evaluation of clustering rules according their complexity for various discretization intervals. The complexity is related to the number of logical conditions in a rule premise. By increasing the number of discretization interval the rules became more complex. If number of intervals is from [90, 110] and rule generation result is bounded, then rule complexity is stable for WINE data. The complexity is greater for unbounded results in comparing with bounded results in many cases.

Fig 6 depicts evaluation of WINE data clustering rules according number of attributes (features) taken into the rule premise. Feature number is general to all rules that include the same attribute in their premise. If two rules include the same attribute, when the general feature number will be 2. Simple rules are easy to read, understand and apply. By increasing the number of discretization intervals the number of features goes down and stops something at the value 10. After removing duplicates the number of distinct features becomes 6. So, the smallest collection of attributes, which describe every available cluster, can be obtained.

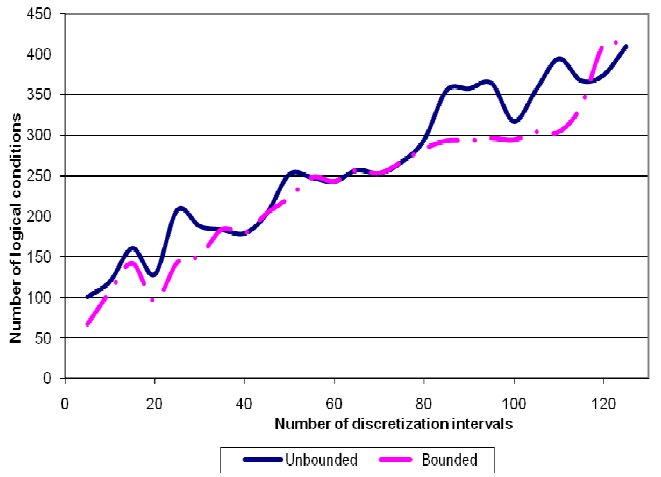

Fig 5. WINE data rules' complexity with bounded and unbounded results

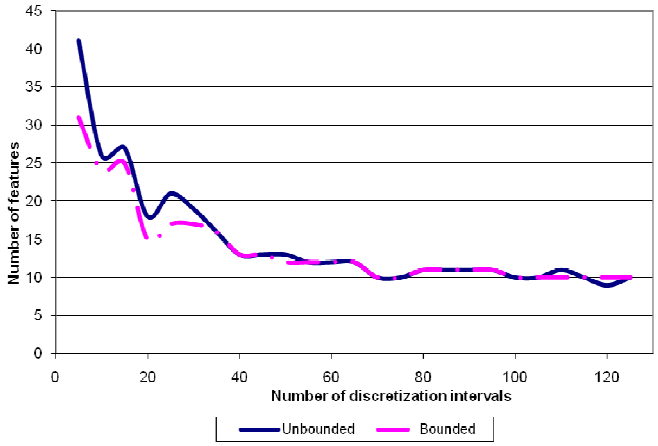

Fig 6. General number of features in rules for WINE data with bounded and unbounded results

The evaluation of number of errors in WINE data when applying clustering rules for the test set having various discretization intervals number is presented in Fig 7. Here, by error we mean the number of examples in POS that are not covered by generated rules, and the number of cases when clustering rules include examples from NEG set to the POS set. By increasing number of discretization intervals, number of errors stabilizes and goes down to 0. The largest number of errors is monitored when clustering results are bounded and comparing small discretization interval is selected. This leads to the larger number of rules do not satisfying given bounds.

Good clustering rules were derived for WINE data with discretization interval number equal to 100. For every cluster one rule was generated:

- The rule describing the first cluster includes two distinct features: feature no.1 and no. 13. The number of logical conditions in is equal to 99.

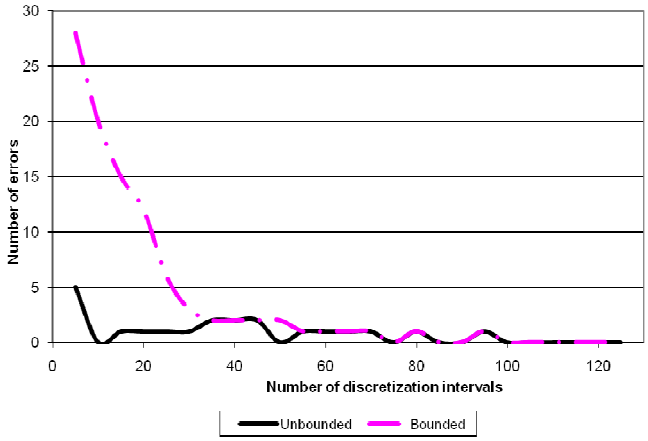

Fig 7. Number of errors in WINE data when applying clustering rules for the test set. Evaluation for bounded and unbounded results

- The rule describing the second cluster includes three distinct features: feature no. 4, no. 9 and no. 13. The number of logical conditions in is equal to 93.
- The rule describing the third cluster includes two distinct features: feature no.7 and no. 10. The number of logical conditions in is equal to 93.

Overall number of logical conditions (99 + 93 + 93 = 285) is smaller in comparing with results of undiscretized data.

Discretization of initial data helps to decrease complexity of clustering rules. The complexity decreases from 500 to 300 logical conditions in average, number of main attributes is also smaller. Increasing number of discretization intervals leads to smaller amount of errors.

IX. Influence of a 'Negative' Data Set to the Clustering Rules

Both CLIP3 and CLIP4 algorithms generate rules according to given POS and NEG data sets. The nodes formed and a final result very mush depending on the negative examples ordering. In this section we summarize investigation results on the influence of a negative data examples ordering to the quality of the clustering rules.

The clustering algorithm generates the next layer nodes according current layer information and the running NEG object. The objects from the NEG set are being taken in the order as they are listed. At least two ways of their selection can be investigated:

1. Selecting one by one in the given order.
2. Random selection.

The first way is not very much attractive, due to its direct impact to the final result of the clustering rules generation process, while the NEG data set is being formed. Depending on the order the objects are listed in the NEG, good or bad results can present.

Random selection of NEG object can also be the reason for both good and bad result. One can look the results obtained randomly selecting unused negative objects and compare those according predefined criteria. So, the desired results can be separated. Such a way of doing is time and computational resources consuming.

Systematically listing the NEG objects can allow us to list negative examples in such a way, that the results obtained would be good enough. But at the moment no clear formal method is described. Yet another way can be in consideration. We can consider the listing of NEG objects, which is built according their influence to the POS data set.

CLIP3 and CLP4 rule derivation algorithms solve minimization problem. During their performance the binary matrixes are built. The construction of a binary matrix is accomplished by following the rules that the value of particular feature of POS object differs from the value of particular feature of running NEG object, then the 1 should be written to the binary matrix. Otherwise, if the values match, the 0 is written to the binary matrix. According to this rule binary matrixes for every NEG observation in NEG data set can be constructed. By summing the ones in such binary matrixes the influence of NEG object to the POS data set is measured. The bigger the sum, lower the influence. Big sum means that the given negative object poorly describes the values in the POS set. According to these measures all NEG objects can be compared.

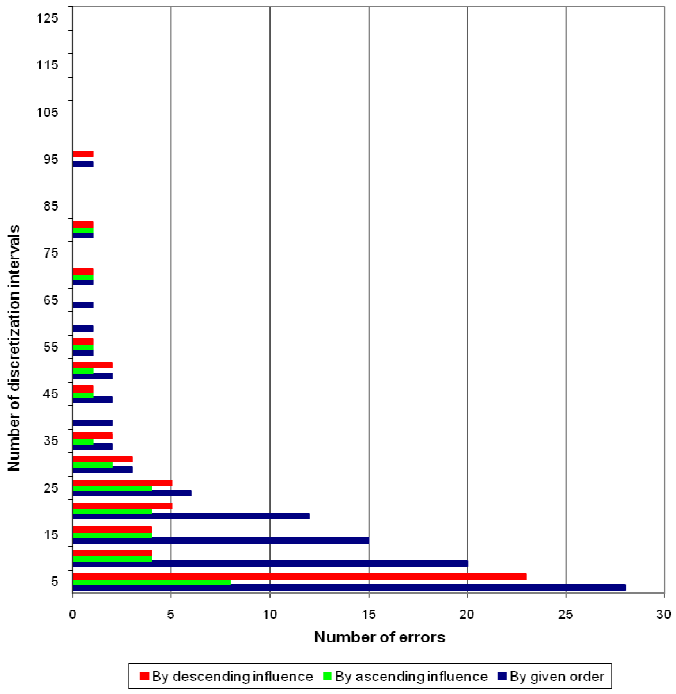

Fig 8. Influence of ordering in NEG set having various discrete intervals in *WINE* data

to the POS. In this case, at first the nodes from the most dissimilar NEG examples in comparing with POS are built in the cluster construction tree.

Fig. 8 depicts influence of ordering in NEG set to the clustering result, while changing the amount of discrete intervals in WINE data. The cases are: negative examples selected according given order, and negative examples sorted according ascending and descending influence to the POS examples. The small amount of errors is when negative examples are given according ascending influence

So, the ordering inside the NEG set influences the final result in clustering rules construction process. Coming from the investigation with WINE data, the smallest number of errors was, when negative examples were ordered according ascending influence to the POS examples. Good results can be also obtained if negative examples are selected randomly.

Concluding, the modified Renyi-CLIPx conceptual clustering algorithm is capable to generate clusters, with the average match to expert judgment of 93.82% having in mind that *WINE* data set was normalized, standard deviation was computed from objects, initial number of clusters was preselected as enough big integer, and objects were redistributed according to the procedure described for in-between cluster redistribution module. Rule derivation module creates not complicated rules, which cover all objects from POS data subset, and improves the quality of rules. The number of logical conditions for *WINE* data set was reduced from 500 to 150.

REFERENCES

[1] D. Baziukaitė, *Learner oriented methods to enhance capabilities of virtual learning environment*, Doctoral Dissertation, Vytautas Magnus University, Kaunas, p. 112, 2007
[2] A. Narščius, J. Golouchova, *Renyi, Iterate ir CLIP3 klasterizacijos algoritmų lyginamoji apžvalga*, Lietuvos pajūrio aplinkos tyrimai, planavimas ir tvarkymas, p. 26-33, 2007
[3] B. Mirkin, *Clustering for Data Mining*, Chapman & Hall/CRC, 2005
[4] K. Cios, D. Wedding, N. Liu, *CLIP3: cover learning using integer programming*, Kybernetes, 26(4-5), 1997
[5] K. Cios, A. Kurgan, *CLIP4: Hybrid inductive machine learning algorithm that generates inequality rules*, Information Sciences, Volume 163, Issues 1-3, p. 37-83, 2004
[6] R. Jenssen, D. Erdogmus, J. Principe, *Clustering using Renyi's Entropy*, IEEE Neural Networks, 523-528, 2003
[7] *WINE data set*. Frank, A. & Asuncion, A. (2010). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
[8] *** Matlab - Statistics Toolbox – Help

Dr. Dalia Baziukė is with the Klaipėda University, Faculty of Science and Mathematics, Department of Computer Science, Herkaus Manto str. 84, LT-92294 Lithuania, email: dalia.baziukaite@ku.lt.

Aleksas Narščius is with the Klaipėda University, Faculty of Science and Mathematics, Department of Computer Science, and Coastal Research and Planning Institute, Herkaus Manto str. 84, LT-92294 Lithuania, email: aleksas.narscius@ku.lt.