Title: Impact of Clustering Parameters on the Efficiency of the Knowledge Mining Process in Rule-based Knowledge Bases

Author: Agnieszka Nowak-Brzezińska, Tomasz Rybotycki

Citation style: Nowak-Brzezińska Agnieszka, Rybotycki Tomasz. (2016). Impact of Clustering Parameters on the Efficiency of the Knowledge Mining Process in Rule-based Knowledge Bases. "Schedae Informaticae" (Vol. 25 (2016), s. 85-101), doi 10.4467/20838476SI.16.007.6188

Schedae Informaticae Vol. 25 (2016): 85–101
doi: 10.4467/20838476SI.16.007.6188

tfml 2017
theoretical foundations
of machine learning, Kraków

Impact of Clustering Parameters on the Efficiency of the Knowledge Mining Process in Rule-based Knowledge Bases

Agnieszka Nowak-Brzezińska[1], Tomasz Rybotycki[2]
[1]University of Silesia, ul. Bankowa 12
40-007 Katowice, Poland,
e-mail: *agnieszka.nowak@us.edu.pl*
[2]IBS PAN, Doctoral Study, ul. Newelska 6
01-447 Warszawa, Poland

Abstract. In this work the subject of the application of clustering as a knowledge extraction method from real-world data is discussed. The authors analyze an influence of different clustering parameters on the quality of the created structure of rules clusters and the efficiency of the knowledge mining process for rules / rules clusters. The goal of the experiments was to measure the impact of clustering parameters on the efficiency of the knowledge mining process in rule-based knowledge bases denoted by the size of the created clusters or the size of the representatives. Some parameters guarantee to produce shorter/longer representatives of the created rules clusters as well as smaller/greater clusters sizes.

Keywords: rule-based knowledge bases, clustering, similarity, visualization

1. Introduction

For the last twenty years, there has been an enormous interest in integrating database and knowledge-based system (KBS) technologies to create an infrastructure for modern advanced applications. The result of it are knowledge bases (KBs) which consist of database systems extended with some kind of knowledge, usually expressed in

Received: 11 December 2016 / Accepted: 30 December 2016

the form of *rules* [1] – logical implications, that can usually be described in form of equation (1):

$$\text{condition}_1 \ \& \ \text{condition}_2 \ \& \ ... \ \& \ \text{condition}_n \implies \text{conclusion}, \tag{1}$$

Such a natural way of knowledge representation makes rules easily understood by experts and knowledge engineers (that are working with KBSs) as well as people not involved in the expert system building (such as data scientists not acquainted with given domain).

1.1. Rules as knowledge representation method

Rules are very specific type of data (knowledge) structure. Usually they are generated from data stored in tabular form (f.e. decision tables). Methods used in order to generate the rules, aims to create so called minimal rules, which means that the rules achieved in this way, have got a short description and cover as many data from the original dataset as possible. Every rule contains two parts: conditional (with at least one premise) and decisional (usually with one conclusion). Sometimes (what makes the analysis more complicated) conclusion of one rule may be a condition in others. In this case it is said that such rules form a chain, and during the inference process they are all processed as a cause and effect chain. Sometimes rules' attributes are weighted therefore the importance of some rules (given as a ordered set of attributes) is higher than others because of difference in weights of their attributes and/or their lengths. What is more, conditional and decisional part of a rule can be also treated differently from each other, f.e. conditional part may have higher priority (greater weights for premises than for a conclusion). All these circumstances make the rules very specific kind of knowledge representation.

1.2. An efficiency of the knowledge mining process

KBs are constantly increasing in volume, thus the knowledge, often stored as a set of rules, is getting progressively more complex and when the rules are not organized into any structure, the KBS isn't as efficient as it could be. There is a growing research interest in searching for methods that could manage large sets of rules using the clustering approach as well as joining and/or reducing rules [2]. Because of many advantages of clustering algorithms [3, 4] it is possible to organize the rules in a smart way. The aim of clustering algorithms is to group the rules into a set of groups (f.e. the hierarchy) of similar rules. To achieve it, some kind of technique, that allows describing similarity between rules, has to be proposed. In the literature there are already a lot of methods of describing similarity between objects, that can be modified to work with rules as well. When two given rules are said to be the most similar (by given similarity measure in a given step of clustering process), they are meant to be clustered before the others. It influences on the further clustering steps. Similarity measure used to find a pair of rules or groups of rules that are the most similar in a given moment is called an *intra-cluster similarity measure* or *inter-object (inter-rule) similarity measure*. The authors studied different intra-cluster similarity measures [5] and choose the following three measures for experimental validation: Gower's measure [6, 7], Simple Matching Coefficient (SMC) [7] and Jaccard's Index [8] sometimes also called weighted similarity or weighted similarity coefficient [7]. They are further described and anylyzed in Section 2.1.1.

The analysis of the rules' similarity can be based on either premises or conclusions of the rules. In this reasearch rules are divided into a number of groups based on similar premises in order to improve the inference process efficiency [9]. This approach is dictated by the forementioned fact, that conditional parts of the rules are generally longer than their decisional parts, and thus making clustering more complex, accurate and interesting. Moreover, the authors directed their research toward the forward (data driven) inference process where the premises of the rules are the basis of searching. To minimize the number of rules that needs to be read before KBS's anwser to given input is reached, instead of searching within the whole set of rules (as in case of traditional inference processes), only representatives of groups would be compared with the set of facts and/or hypothesis to be proved. The most relevant group of rules is selected and the exhaustive searching is done only within this group. This way, given a set of rules, new knowledge may be derived using a standard forward chaining inference process, which can be described as follows: each cycle of deductions starts with matching the condition part of each rule with known facts. If at least one rule matches the facts asserted into the rule base, it is fired. It's really crucial to find and describe all the factors which influence clustering results and interference efficiency as it'd help in designing or partitioning of KBs in order to maximize KBS's effectiveness.

There is also the other type of cluster similarity measure – so called *inter-cluster* similarity measure. It's used to describe how much each group resembles one another basing on their members (see the details in Section 2.1.2.).

Examining the influence of choosing different similarity measures on the efficiency of clustering algorithms, is the main goal of this research. It's crucial to answer the question if a given similarity measure influences the shape of grouped KB's structure. To have a chance to analyze it, the silimarity measures described in section 2.1. were implemented by the authors. The result of this implementation is the `CluVis` system, described briefly in Section 4. The system allows to examine different similarity measures and methods of hierarchical clustering for any given knowledge base in predefined form described further in [10].

The rest of the paper is organized as follows. We first mention all related efforts in the study of rules clustering in Section 2., where the description of different inter and intra cluster similarity measures were described. Section 3. focuses on two selected visualization algorithms designed for hierarchical strucures. Short description of the authors system `CluVis` can be found in Section 4., whereas Section 5. describes experimental setup, evaluation methodology and the results on public data sets.

2. Rules clustering

Nowadays mankind's knowledge is growing rapidly. People are constantly developing new ways of using this knowledge in practice. As a result KBSs came into being – decision systems, in particular, are prime exmample of that. These systems, often limited to only one domain, tend to store their knowledge in special form. Rule-based knowledge bases (as mentioned in 1.) are most common, because it's easier (for knowledge engineers and experts) to present laws and rudiments of given domain in this shape. Rules stored in knowledge bases are often unorganized as usually there's no reasonable way of doing so, because it's hard (if not impossible) to judge eg. which law or theorem is more important than the other. However, to ensure KBSs' optimal work, it'd be necessary to decrease amount of data that system needs to read to get final conclusion. One way to achive that is proposing some kind of rules partitioning that'd agregate similiar and separate different rules, describing each of such groups with proper description (generalization of it's members) and thus allowing searching through KB from most general groups to specific rule and, ipso facto, successfully decreasing amount of data read during generating KBS's anwser. The same approach may be neccessary during updating KB (deleting redundant rules etc.), where searching through KB is also required. It's possible to find such partitioning using data exploration methods. The most natural approach seems to be clustering.

Clustering is one of the oldest and most common methods of organizing data sets. The goal of clustering is to maximize intra-cluster similarity and minimize inter-cluster similarity, thus creating a partition, where similiar rules are joined into one cluster and rules different from others are singled out. During this unsupervised process similiar objects (according to given similarity measure) are joined into groups thus often decreasing amount of data required to be analyzed in order to extract knowledge from examined data set. Several methods of clustering has been proposed, each of them includes variety of different techniques. Due to forementioned fact selecting proper algorithm is a nontrivial task as there are many factors to be considered. In this work, sets of complex objects known as rules (denoted as Horn's clauses) are being examined. As the authors were aiming to find hierarchy-like partitioning that'd ensure that finding specific rule is achieved through searching through more general groups, hierarchical clustering methods were used. Hierarchical clustering has also one more advantage. It doesn't impose on any special methods of describing clusters similarity and thus can be used for rules clustering without additional modifications. To avoid increasing complexity of research the authors selected one of the most well known algorithms from this popular group of techniques – Classical AHC algorithm – and used to organize several KBs.

2.1. Similarity measures

Measuring similarity or distance between two data points is a core requirement for several data mining and knowledge discovery tasks that involve distance computation.

The notion of similarity or distance for categorical data is not as straightforward as for continuous data. When data consists of objects that agregate both types at once the problem is much more complicated. It is necessary to find a measure, that could deal with this case.

2.1.1. Intra-cluster similarity measures

In this paper, the authors study a variety of similarity measures. Having a set of attributes A and set of sets of their values $V = \bigcup_i V_i$ rules premises and conclusions are build using pairs (a_i, v_j), $a_i \in A, v_j \in V_i$ called descriptors D as a vector of lenght equal to number of attributes in permise and conclusion of given rule, where i-th position denotes the value of i-th attribute for a given premise/conclusion attribute.

In all similarity measures, described in this work, similarity S between two rules r_i and r_j can be denoted as weighted sum of similarities s_k considering k-th common attribute of these rules. This can be written as equation (2):

$$S(r_i, r_j) = \sum_{k: a_k \in (A(r_i) \cap A(r_j))} w_k s_k(r_{ik}, r_{jk}), \tag{2}$$

where $A(r)$ is set of attributes of rule r, w_k is the weight assigned to k-th attribute and r_{ik} and r_{jk} are values of k-th attributes of i-th and j-th rule respectively.

SMC (simple matching coefficient) [7] is the simplest measure of similarity considered in this work. It handles continous attributes the same way it does with categorical attributes, namely (equation (3)):

$$s(r_{ik}, r_{jk}) = \begin{cases} 1 & \text{if } r_{ik} = r_{jk} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

In this case, however, overall similarity of rules S is simple sum, as weight w_k of each attribute is equal to 1. Due to that fact this similarity measure tends to favor rules with more attributes. More advanced form of this measure is Jaccard index [8, 7]. It eliminates forementioned drawback of SMC by setting weight $w_k = \frac{1}{Card(A(r_i) \cup A(r_j))}$, where $Card : V \to \mathbb{N}$ is the cardinality of a set.

Last measure described in this work is Gower's index [6]. This measure is most complicated one, that the authors have used, as it handles categorical data differently from numeriacal data. Similarity considering categorical data is count the same way as in case of Jaccard or SMC. Similarity of continous attributes can be denoted as following (equation (4)):

$$s(r_{ik}, r_{jk}) = \begin{cases} 1 - \frac{|r_{ik} - r_{jk}|}{range(a_k)} & \text{if } range(a_k) \neq 0 \\ 1 & \text{otherwise} \end{cases} \tag{4}$$

where $range(a_k) = max(a_k) - min(a_k)$ is range of k-th common attribute.

All measures described in this subsection were used as a parameter of classical AHC algorithm in order to examine influence they have on resultant structure of clustering. It was shown that f.e. some of them tends to generate structures with larger number of ungrouped rules and that different inter-object similarity measures influences average length of cluster representative, thus allowing one to consider KB partitioning basing on cluster representatives length.

2.1.2. Inter-cluster similarity measures

Apart from the usual choice of similarity functions, linkage criterion must be selected (since a cluster consists of multiple objects, there are multiple candidates to compute the distance to). There are many possible ways of describing inter-cluster similarity. The most popular among them are known as Single Link (SL), Complete Link (CoL), Average Link (AL) and Centroid Link (CL) [3, 4, 11].

Single link is the most head-on approach, as it describes similarity of the clusters basing on their most similar objects, thus in case of rules-clustering one may say, that clusters of rules are as similar as their most common two rules. This method is known to cause undesireable occurence called cluster chaining, wherein long clusters are being created. In general, however, it's not proper partitioning for given dataset and thus another method should be proposed.

The most similiar to SL is method called Complete Link. Both of these methods returns similarity of single pair of rules as similarity of two clusters, however, in case of complete link, clusters are only as similar as their two most disrinct rules. The values of CoL are obviously lower (on average) than in case of SL. It is known in the literature (and also had been shown during experiments in this work) that CoL tends to generate partitionings wherein there's lower number of small groups and bigger number of larger ones. Both of forementioned methods share a common drawback, namely: they are sensitive to noisy data. The reason for that is that they both base their result on single pair of rules instead of considering whole contents of the clusters.

Another inter-cluster similarity measure that was used in this work is Average Link. Let C_i, C_j be two clusters of rules. Then similarity between C_i and C_j can be defined by equation (5):

$$AL(C_i, C_j) = \frac{\sum_{r_i \in C_i} \sum_{r_j \in C_j} S(r_i, r_j)}{\overline{\overline{C_i}} * \overline{\overline{C_j}}} \tag{5}$$

In other words AL is described as mean similarity of all rules inside examined clusters. It's sensitive neither to noisy data nor to cluster chaining as it considers all rules that are inside given clusters.

Another measure that considers all rules within given clusters is centroid link. This one however, instead of considering similarity between rules that are inside examined clusters, considers only similarity between two virtual objects, called centroids of clusters. Usually centroids are described as geometrical center of the cluster, however, in case of rules, defining geometrical center of cluster is non-trivial task, thus representative of this cluster was used instead. It was dictated by assumption that representative of the cluster (further described in subsection 2.2.) is meant to be the most general rule in the cluster – one may say most ,, centered ” one. Let $c(C_i)$ be centroid of i-th cluster. Then similarity between clusters C_i and C_j can simply be defined as $CL(C_i, C_j) = S(c(C_i), c(C_j))$.

In previous authors' researches it was noticed that the method of creating the representatives are equally important with the clustering algorithm and similarity measures used to clustering rules. In this work, further researches on this matter were conducted.

2.2. Cluster's representative

It is very important for data to be presented in the most friendly way. Sole visualization of clustering (described further in 3.) is not enough, as it would only reduce the whole pattern discovery to examining an accumulation of shapes, thus some kind of symbolic description has to be proposed. Cluster representatives are the proposed solution for this issue. There are many methods of creating representatives. In this work representative aims to be an average rule of the cluster, basing on cluster's content. Its creation algorithm can be described as follows. Having as input data: cluster C, and a threshold value t [%], find in the cluster's attributes set A only these attributes that can be found in t rules and put them in set A'. Then for each attribute $a \in A'$ check if a is symbolic or numeric. If it is symbolic then count modal value, if numeric - average value and return the attribute-value pairs from A' as cluster's representative. The representative created this way contains attributes that are the most important for the whole cluster (they are common enough). This way the examined group is well described by the minimal number of variable[1]. It should be possible to characterize the clusters using a small number of variables (the number of attributes attained by this method strongly depends on selected threshold value). It is very important to examine the quality of created clusters and to generate the well-formed descriptions for them, what is difficult especially when the objects of clustering are rules.

2.3. Clustering algorithm

As mentioned in section 2. Classical AHC algorithm was used to for rules clustering in this work. It's essential to point out that during each iteration step it joins only two most similiar clusters (not all clusters with similarity exceeding given threshold). Decision of choosing this algorithm over general AHC was dictated by the fact, that selecting rational threshold value, especially when grouping complex objects such as rules, would require deep analysis of each examined data set.

[1] However the authors see the necessity to analyze the more methods for creating clusters' representatives and their influence on the resultant structure efficiency.

Clustering algorithm used in this work agregates all the elements forementioned in this section. As AHC algorithm doesn't specify neither inter-object nor inter-cluster similarity measure, they have to be precised as algorithm's parameters. In this work all measures specified in 2.1.1. and 2.1.2. were used in all possible configurations. During each merging step created cluster is given a description, in form of representative, generated in a way described in 2.2. It's worth mentioning that representative is created basing solely on rules within given cluster, not on representative of cluster figuring higher in hierarchy.

The grouping is complete after selected number of groups is reached. The number of produced clusters is in range from 1 to N, where N is the number of rules in examined knowledge base. After grouping is finished resultant structure can be visualized using algorithms described further in 3.

3. Knowledge base visualization

The most common way of clustering presentation is tree like structure called dendrogram, however, it has some fatal flaws that makes it inadequate for researches such as presented in this work. Among others, the most vital issue is that it isn't suited for representing large clusterings, as it quickly becomes less readable. This problem concerns many visualization algorithms, but some of them loses their readability much slower e.g. treemaps. In this work two treemap algorithms were used – rectangular treemap [12] and circular treemap [13]. The differences between the two are very distinct for each of them base on different geometrical shapes and has different methods of deploying them. In this work classical slice-and-dice deployment method was used for rectangles as proposed in [12] and method described in [10] was used for circles.

Figure 1 presents the case of using the `CluVis` to cluster 42 rules in a given knowledge base with 70 attributes into 10 groups. Clusters are presented graphically using Circular Treemap (as selected visualization algorithm) and classical AHC (as clustering algorithm) with Gower measure for inter-cluster similarity measure and (CL) Complete link as intra cluster similarity measure. For this particular case the biggest cluster's size is 18 rules (which is 42% of all rules in KB), (denoted as $J33$) with the representative's length equal to 67 descriptors (which is also the maximum length for representatives in general). The other clusters contains the following number of rules: $4, 4, 6, 1, 1, 1, 4, 2, 1$.

Visualizations generated using these algorithms, without any additional features, would only be accumulations of shapes. In some cases such solution would seem sufficient, however, in general, especially when exploring large sets of rules, it is confusing or chaotic. To make exploration of large KBs easier each visualization is responsive. Each shape stores information about cluster it represents. Moreover there is possibility of future examining objects agregated in cluster by selecting it. To ensure that desired cluster is selected, active shape is highlighted. To make visualization even more readable, colors (of the rainbow) were used to mark procentage size of the cluster. Colors and their order were selected in such way to make it easy to remember.

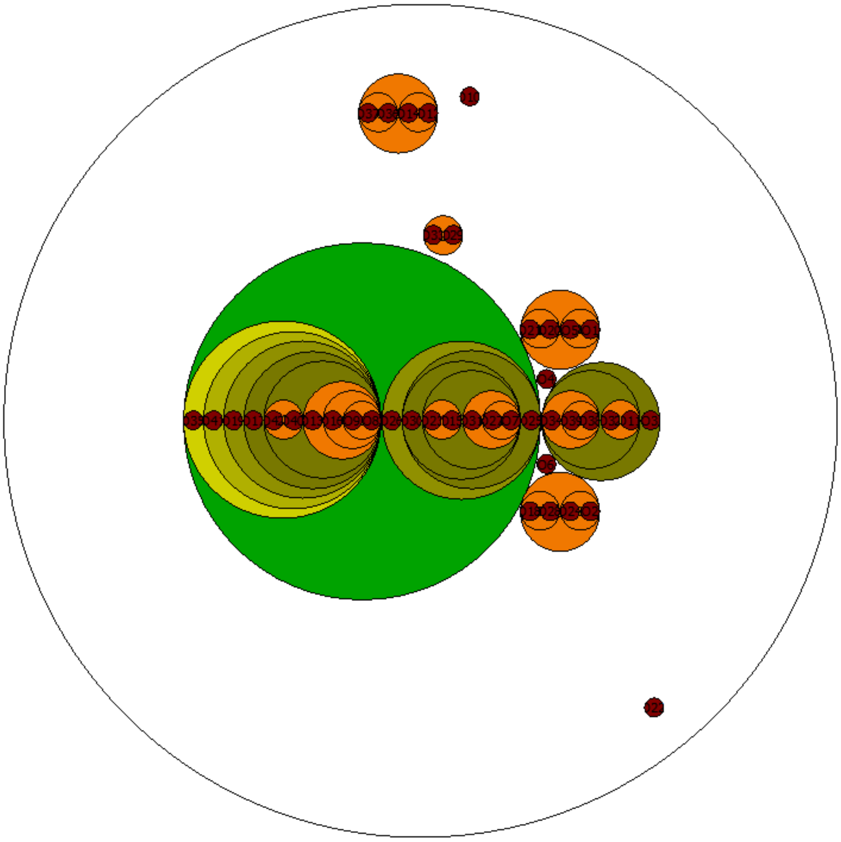


Figure 1. Sample treemap visualization.

4. CluVis

`CluVis` [10] is an application designed to group sets of rules generated by $RSES$ [14] and visualizing them using selected treemap methods. It is first application capable of working on raw KBs as generated from $RSES$. It was successfully used in previous researches to group and visualize medical knowledge bases generated from artificial data sets available on [15] as well as one generated from real medical data [11]. Moreover, it aggregates functionalities of both clustering and visualization software, making it universal tool for exploring KBs. Along its main functionalities (many of which can be seen in figure 2), described in more detail in sections 2. and 3. `CluVis` is capable of generating reports of grouping (to `txt` or special `xml` files which can be opened in such applications as e.g. `Libre Calc`) containing detailed information about each obtained cluster and about clustering in general (number of nodes, min/max/avg representative length...). It's also possible to save generated visualization to `png` file as well as to find best clustering (according to implemented cluster validation indexes – MDI and $MDBI$ – not discussed in this article). `CluVis` is an open source application written in C++11 using QT graphic libraries. It's available in english and polish and its source code can be downloaded from https://github.com/Tomev/CluVis.

Every XML report file, generated using the authors application, contains the following informations: index of an experiment (`Index`), the name of the knowledge base file (`Name of the base`), the number of attributes (`Attributes number`), of objects (`Objects number`), of nodes in the created hierarchy (`Nodes number`), of created clusters (`Clusters number`). It also says what was the algorithm used to visualize the groups of rules (X or Y), which clustering algorithm it uses, what was the intra-cluster and inter-cluster similarity measure, what part of the rule (conditional or/and decisional) was analyzed as well as the names of the biggest and the smallest cluster (`The biggest cluster` / `The smallest cluster`). Moreover it gives the values for: `Coverage sum`,
`Biggest cluster size`, `Biggest representative length`, `Ungrouped objects`,
`Biggest representative size`, `Smallest representative size`,
`Average representative size`, `MDI\verb`, `MDBI`. It also contains the details data of each cluster, like `Index`, `Cluster's name`, `Cluster's size`,
`Objects percent in cluster`, `Cluster's nodes number`,
`Cluster's nodes percent`, `Cluster's coverage`, `Cluster's coverage percent`,
`Cluster's representative`, `Cluster's representative length`.

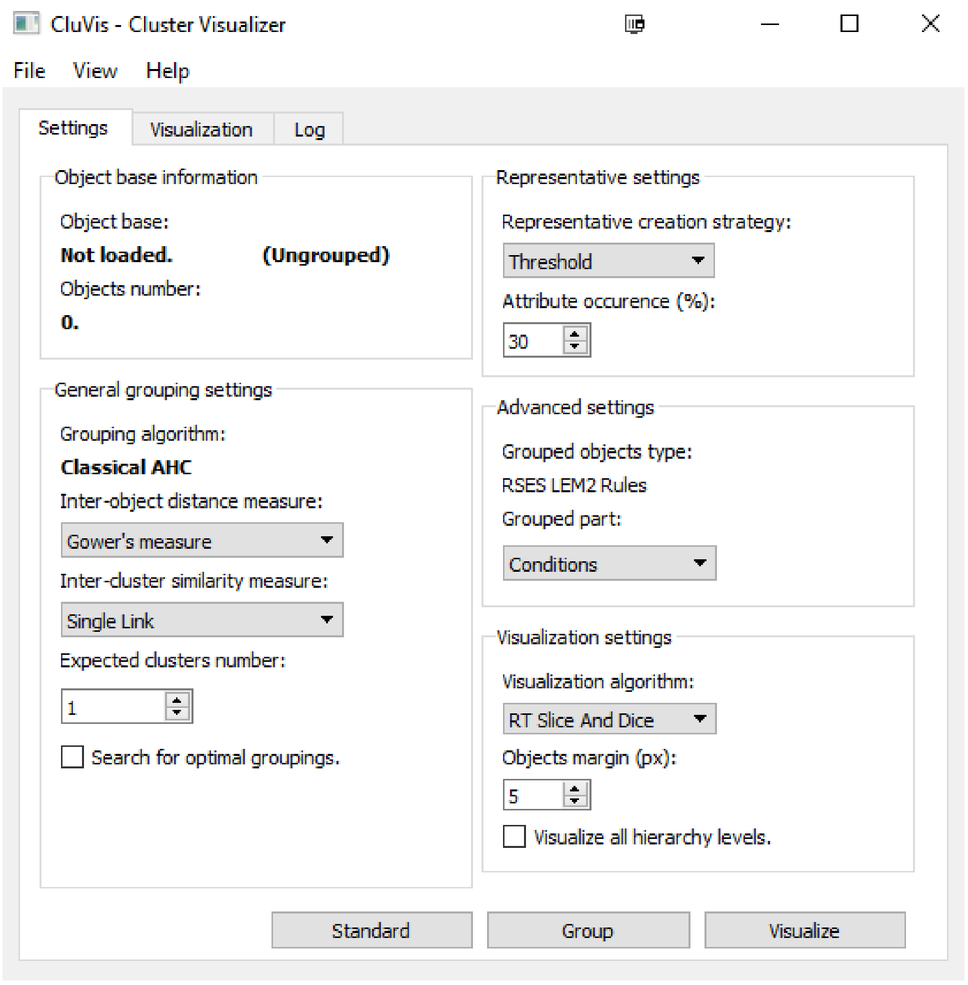
Then an example of the cluster representative (which contains 25 of descriptors) may looks like:

```
(history_ringing=f)&(history_fullness=f)&(m_m_sn_gt_500=f)&
(m_s_sn_gt_2k=f)&(boneAbnormal=f)&(history_buzzing=f)&(m_m_gt_2k=f)&
(m_gt_1k=f)&(history_dizziness=f)&(m_s_sn_gt_1k=f)&(airBoneGap=f)&
(history_fluctuating=f)&(m_s_gt_500=f)&(age_gt_60=t)&(air=mild)&
(history_noise=t)&(m_p_sn_gt_2k=f)&(m_m_sn_gt_2k=f)&(history_heredity=f)&
(history_recruitment=f)&(late_wave_poor=f)&(m_at_2k=f)&(m_cond_lt_1k=f)&
(bser=MISSING)&(m_s_sn_gt_3k=f)=>(class=cochlear_age_and_noise)
```

Automatically created KBs (e.g. creating rules from dataset using $LEM2$ algorithm) have a chance to contain some undesired rules (like ones that would never be activated or are simply redundant). It is essential to maintain simplicity of KBs thus some method of eliminating these rules is required. As $CluVis$ is used to transform unorganized KBs into organized ones, presented in form of responsive visualization that enhances readability of KB, it's a perfect tool for this task.

The process of organizing knowledge bases in `CluVis` is as follows. After importing KB into the application and selecting clustering parameters grouping can be performed. First selected file is scanned to see weather it has proper format. In the next step data about attributes is gathered (such information as frequency of each value of each categorical attribute, maximal and minimal values of continous attributes...) as they are used in some similarity measures. Then rules are transformed from lines of text into hashmaps, which are basically vectors[2], which are stored as a singular clusters. Then similarity matrix (which is triangle matrix holding information about similarity between i-th and j-th cluster) is built using similarity measure selected during first phase. Then two the most similiar clusters are joined, and the

[2] The i-th variable value is accessed by it's name in map, not by it's index


CluVis - Cluster Visualizer

File View Help

Settings | Visualization | Log

Object base information

Object base:

Not loaded. **(Ungrouped)**

Objects number:

0.

General grouping settings

Grouping algorithm:

Classical AHC

Inter-object distance measure: Gower's measure

Inter-cluster similarity measure: Single Link

Expected clusters number: 1

☐ Search for optimal groupings.

Representative settings

Representative creation strategy: Threshold

Attribute occurence (%): 30

Advanced settings

Grouped objects type:

RSES LEM2 Rules

Grouped part: Conditions

Visualization settings

Visualization algorithm: RT Slice And Dice

Objects margin (px): 5

☐ Visualize all hierarchy levels.

Standard | Group | Visualize

Figure 2. CluVis's GUI.

similarity matrix is updated. During merging of two smaller clusters, their representative is calculated. There may be many possible stop conditions for AHC (e.g. end when similarity of most similiar objects is equal to 0). `CluVis` ends clustering after given (as clustering parameter) number of clusters is reached. The grouping ends when stop contidion is satisfied (selected number of clusters is formed). As soon as grouping is complete visualization can be generated.

Graphical representation of clustering can be performed in two ways – with fully hierarchical view, or not. Visualization is responsive meaning that selecting a cluster and clicking it with right mouse button will generate new visualization representing internal structure of selected cluster. Considering that during each step of used AHC algorithm only two objects are merged usually structure of internals is trivial (large cluster + single object), however, it's nonetheless a method of exploring KB by diving deeper into it. To ensure proper direction of diving, one may generate a report from each cluster on current visualization and select one that seems most appropriate. Sample visualization was presented on figure 1.

5. Experiments

In this section, an experimental evaluation of 3 similarity measures and 4 clustering methods on 7 different KBs [15] is presented. Decision rules were generated from the original data using `RSES` software and $LEM2$ algorithm [14]. The smallest number of attributes was 5, the greatest 280. The smallest number of rules was 42, the greatest 490.

Table 1. Characteristics of the experimental data.

	AttrN	RulesN	ClustersN	UngroupedN
Arythmia	280	154	12,5±2,52	5,78±5,68
Audiology	70	42	6,92±3,02	3,56±2,85
Autos	26	60	7,89±2,25	3,57±2,92
Balance	5	278	19±9,00	7,46±8,70
Breast	10	125	11±1,01	5,08±3,35
Diab	9	483	28,74±19,13	11,75±13,75
Diabetes	9	490	29,5±19,64	13,58±14,87

Table 2. Data gathered during experiments.

	BRS	ARS	WARS	BRL	BCS
Arythmia	151,9±4,6	133,4±11,6	2,1±0,2	147,4±1,5	111,5±41,7
Audiology	67,0±0,4	49,7±1,2	1,5±0,4	66,8±0,5	29,8±7,8
Autos	11,9±1,9	8,6±1,7	3,2±0,6	10,7±0,5	38,3±14,0
Balance	4±0	3,5±0,4	1,4±0,2	4±0	180,2±93,5
Breast	9±0	7,1±1,1	1,4±0,2	9±0	7,6±32,8
Diab	5,4±0,5	3,4±0,8	2,9±0,7	4,8±0,6	314,8±137,2
Diabetes	5,6±0,7	3,3±0,8	2,9±0,7	4,9±0,3	335,2±140,7

All the details of analyzed datasets are included in table 1 and 2. The meaning of the columns in table 1 and 2 is following:

$AttrN$ – number of different attributes occuring in permises or conclusions of rules in given knowledge base.

$RulesN$ – number of rules in examined knowledge base.

$ClustersN$ – number of nodes in dendrogram representing resultant structure.

UngroupedN – number of singular clusters in resultant structure of grouping.

BRS – Biggest representative size – number of descriptors used to describe longest representative.

ARS – Average representative size – average number of descriptors used to describe cluster's representatives.

WARS – Weighted Average representative size (AttrNumber) – division of average number of descriptors used to describe cluster's representative in give data set and number of attributes in this data set.

BRL – Biggest representative length – number of descriptors in biggest cluster's representative.

BCS – Biggest cluster size – number of rules in the cluster that contains the most of them.

The performance of different similarity measures was evaluated in the context of knowledge mining using informations like: the number of rules clusters (*CN* – Clusters number), the number of ungrouped rules (*U* – Ungrouped objects), the sizes of the biggest cluster (*BiggCluS* – Biggest cluster size) as well as its representative (*BiggRepS* – Biggest representative size) and the representative the most specific (*BiggRepL* – Biggest representative length). More specific means more detailed, containing a higher number of descriptors.

The optimal structure of *KB*s with rules clusters should contain the well separated groups of rules, and the number of such groups should not be too high. Moreover, the number of ungrouped rules should be minimal. Creating an optimal description of each cluster (representative) is very important because they are used further to select a proper group (and reject all the others) in inference process, in order to mine knowledge hidden in rules (by accepting the conclusion of the given rule as a true fact). The experimental results verifies the initial hypotheses about the inter and intra cluster similarity measures. As can be seen from Tables 3 and 4 no single measure is always superior or inferior. This is obvious since each *KB* has different characteristics (different number of attributes and/or rules) as well as different types of attributes. The use of some measures however, guarantees achieving more general or more specific representatives for created rules clusters. There are some pairs of measures that exhibit complementary performance, i.e. one performs well where the others perform poorly and vice-versa.

Table 3. Influence of inter-cluster similarity measures on respective values.

	CN	BiggCluS	BiggRepL	U	BiggRepS
Gower	16,6±14,1	157,2±147,4	35,7±51,2	8,1±9,9	36,4±52,3
SMC	16,4±14,3	155,6±143,6	35,4±50,8	5,6±6,7	36,3±52,4
WSMC	22,1±16,5	211,5±152,8	5,7±2,0	6,1±9,1	6,0±1,9

Table 3 and figure 3 show that choosing different intra-cluster similarity measures does not influence the overall efficiency (the exception is Jaccard's index).

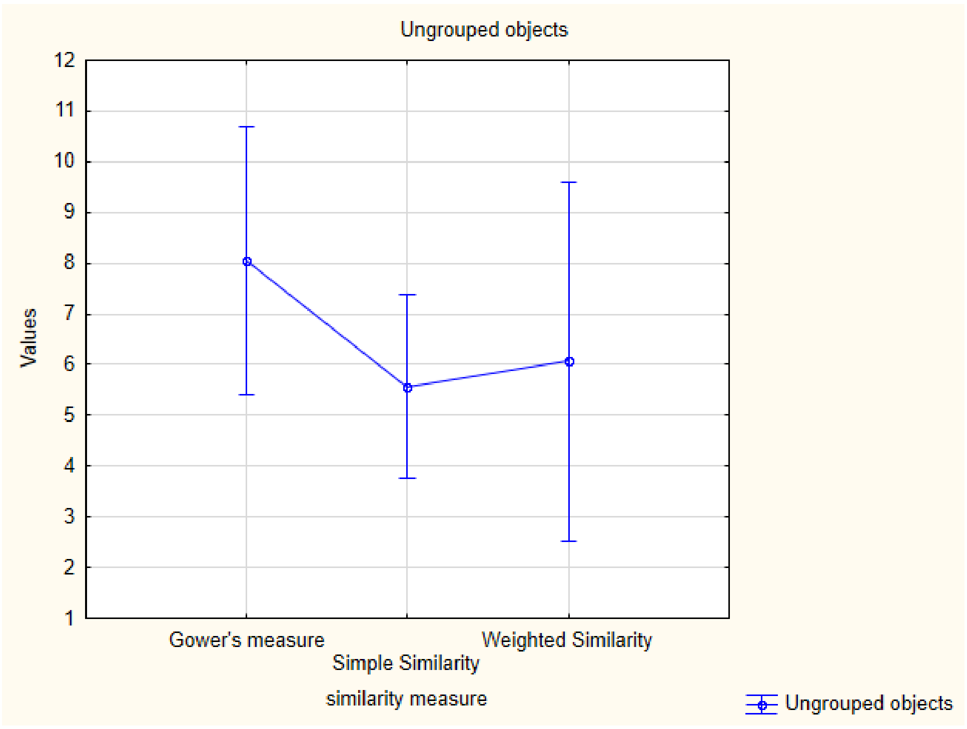


Figure 3. Ungrouped rules – interval plot for inter-cluster similarity measures.

Table 4 (with figure 4) however shows that the size of the biggest cluster and the number of ungruped rules depends on the inter-cluster similarity measures. The experiments confirmed that the SL method produces straggling clusters, called chaining, where clusters may be forced together due to single elements being close to each other, even though many of the elements in each cluster may be very distant from each other. CoL, on the other hand, tends to find compact clusters.

Table 4. Influence of intra-cluster similarity measures on respective values.

	SL	CL	AL	CoL
CN	16,5±14,2	16,5±14,1	16,6±14,1	16,6±14,1
U	12,1±11,8	2,1±3,1	4,5±5,2	10,4±10,9
BigCluS	213,5±166,2	84,0±89,8	157,2±139,8	167,8±142,6
BigRepL	35,4±50,4	35,1±50,6	35,4±50,4	35,4±50,5
BigRepS	36,0±51,8	36,4±51,5	36,5±51,5	36,5±52,2

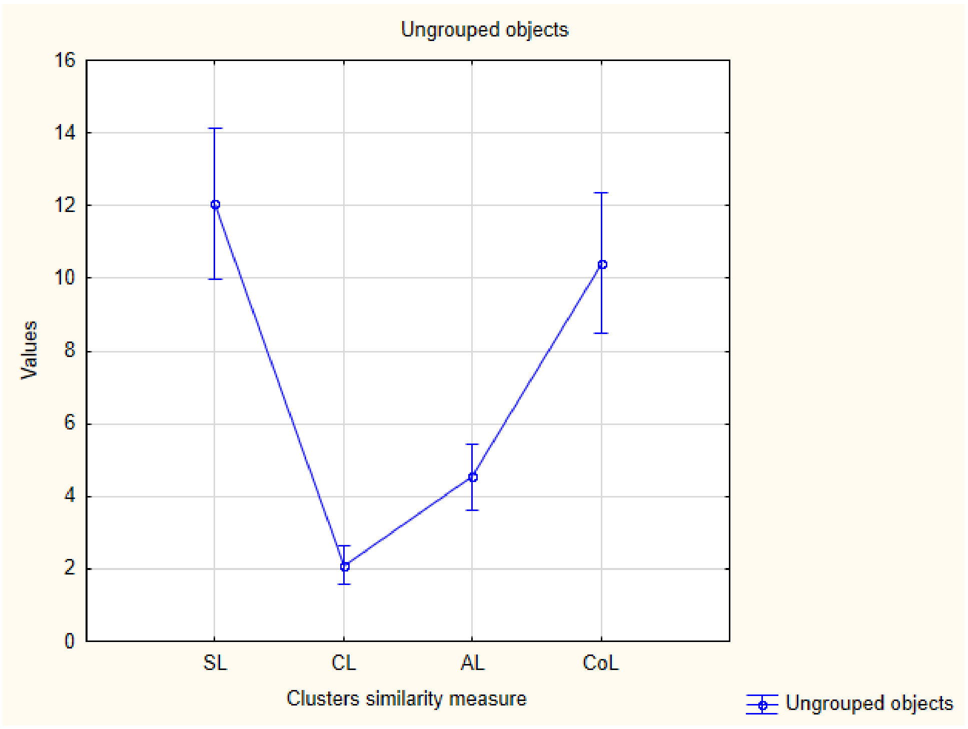


Figure 4. Ungrouped rules – interval plot for intra-cluster similarity measures.

6. Summary

This article presents how exploration of complex KBs can be performed using clustering and visualization of rules clusters and presents the application of clustering as a knowledge extraction method from real-world data. Clustering a large set of objects (rules in this case) is not enough when exploring such an enormous amount of data in order to find some hidden knowledge in it. The extraction of valuable knowledge from large data sets can be difficult or even impossible. Modularization of KBs (by clustering) helps to manage the domain knowledge stored in systems using the described method of knowledge representation because it divides rules into groups of similar forms, context, etc. The authors analyze an influence of different clustering parameters on the quality of created structure of rules clusters and the efficiency of the knowledge mining process for rules / rules clusters. In the course of the experiments, three different similarity measures and four clustering measures have been examined in order to verify their impact on the size of the created clusters and the size of the representatives. The experiments have revealed that there is a corelation between the parameters used in the clustering process and future efficiency levels of the knowledge mined from such structures: some parameters guarantee to produce shorter/longer representatives of the created rules clusters as well as smaller/greater clusters sizes.

The authors propose to use clusters of rules and visualize them using treemap algorithms and hope that this two-phase way of rules representation allows the domain experts to explore the knowledge hidden in these rules quicker and more efficiently than before. In the future, the authors plan to extend the software's functionality, especially in the context of parameters used in clustering and visualizing procedures, as well as importing other types of data sources. It would be easier then to support human experts in their everyday work by using the created software (`CluVis`) in work with many expert systems.

7. References

[1] Mulawka J.J., *Systemy Ekspertowe.* Wydawnictwo Naukowo-Techniczne, Warszawa, 1996.

[2] Latkowski R., Mikołajczyk M., *Data decomposition and decision rule joining for classification of data with missing values.* In: *Rough Sets and Current Trends in Computing.* vol. 3066 of *Lecture Notes in Computer Science.*, Springer Berlin Heidelberg, 2004, pp. 254–263.

[3] Morzy T., *Eksploracja danych. Metody i algorytmy.* Wydawnictwo Naukowe PWN, 2013.

[4] Wierzchoń S.T., Kłopotek M.A., *Algorithms of Cluster Analysis.* Wydawnictwo IPI PAN, Warszawa, 2015.

[5] Boriah S., Chandola V., Kumar V., *Similarity measures for categorical data: A comparative evaluation.* In: Chid Apte, Haesun Park K.W., Zaki M.J., eds.: *Proceedings of the 2008 SIAM International Conference on Data Minning*, Society for Industrial and Applied Mathematics, 2008, pp. 243–254.

[6] Gower J.C., *A general coefficient of similarity and some of its properties.* Biometrics, 1971, 27, pp. 857–871.

[7] Nowak-Brzezińska A., Jach T., *Wnioskowanie w systemach z wiedzą niepewną.* In: *Studia Informatica.* vol. 32 No 2A. Wydawnictwo Politechniki Śląskiej 2011, pp. 377–389.

[8] Jaccard P., *tude comparative de la distribution florale dans une portion des alpes et des jura.* Bulletin de la Socit Vaudoise des Sciences Naturelles, 1901, 37, pp. 547–579.

[9] Nowak-Brzezińska A., *Mining rule-based knowledge bases inspired by rough set theory.* 2016, 148 (no. 1–2), pp. 35–50.

[10] Rybotycki, T., *Visualization of hierarchical structures in rule-based knowledge bases.* March 2015.

[11] Nowak-Brzezińska, A., Rybotycki T., *Visualization of medical rule-based knowledge bases.* Journal of Medical Informatics & Technologies, 2015, 24, pp. 91–98.

[12] Shneiderman B., *Tree visualization with tree-maps: 2-d space-filling approach.* 1992, 11, pp. 92–99.

[13] Wetzel K., *Pebbles – using circular treemaps to visualize disk usage.* http://lip.sourceforge.net/ctreemap.html, 2004.

[14] Bazan J.G., Szczuka M.S., Wroblewski J., *A new version of rough set exploration system.* In: *Rough Sets and Current Trends in Computing.* vol. 2475 of *Lecture Notes in Computer Science.*, Springer Berlin Heidelberg, 2002, pp. 397–404.

[15] Lichman M., *Machine learning repository.* http://archive.ics.uci.edu/ml, 2013 Accessed in October 2016.