JOURNAL OF APPLIED
COMPUTER SCIENCE
Vol. 25 No. 2 (2017), pp. 7-27

Outlier Mining in Rule-Based Knowledge Bases

Agnieszka Nowak-Brzezińska[1]

[1] *Institute of Computer Science*
Silesian University
Bankowa 12, 40-007 Katowice, Poland
agnieszka.nowak@us.edu.pl

Abstract. *This paper introduces an approach to outlier mining in the context of rule-based knowledge bases. Rules in knowledge bases are a very specific type of data representation and it is necessary to analyze them carefully, especially when they differ from each other. The goal of the paper is to analyze the influence of using different similarity measures and clustering methods on the number of outliers discovered during the mining process. The results of the experiments are presented in Section 6 in order to discuss the significance of the analyzed parameters.*

Keywords: *outlier detection, similarity analysis, clustering, knowledge-based systems..*

1. Introduction

Outlier detection is a fundamental issue in data mining, it has been specifically used to detect and remove anomalous objects from data. Data mining, in general, deals with the discovery of nontrivial, hidden and interesting knowledge from different types of data. With the development of information technologies, the number of databases and their dimensions and complexity grow rapidly. One of the basic problems of data mining is outlier detection. The identification of an outlier is affected by various factors, many of which have become the subject of practical applications such as public health or credit card transactions. In the first case (public health), outlier detection techniques help to detect anomalous patterns in patients' medical data which could be symptoms of an ailment. Generally, outliers are the points which are different from or inconsistent with the rest of the data. It can be novel, new, abnormal, unusual or noisy information thus it is often more interesting than the majority of the actual data. One of the very efficient technique of data mining process, which allows to discover outliers, is clustering. It works by grouping the observed data into clusters, according to a given similarity or distance measure (details are included in Section 4.2). Usually, for every cluster its representative point is selected and then, each new data point is classified as belonging to a given cluster according to the proximity to the corresponding representative point [1]. Points that do not belong to any cluster are named outliers and represent the anomalies in the detection process. There are different type of data to be analyzed: text, DNA sequences, numerical data, mixed type data, pictures, videos etc. Sometime the data may have represent some very specific format. The examples of such data are rules (logical sentences in the form of Horn's clause: `if` $cond_1 \& cond_2 \& \ldots \& cond_n$ `then` $conclusion$)[2] [1]. More details about the rule-based knowledge representation can be found in Section 2. The goal of this research is the issue of outliers' occurrence in rules. Thus, the main goal of this paper is to present recent approaches to outlier mining in rule-based knowledge bases (*KBs*). Such data sources are very popular in the area of decision support systems (*DSS*). It is very important to have the possibility to explore the *KBs*, especially because they often consist of many correlations, dependancies or even unusual cases (rules). A few years back, outlier detection was just one of several steps of data preparation procedure. Frequently, the data that have been denoted as outliers would have been removed from the dataset and considered as errors. Nowadays outliers are no longer (or at least not entirely) seen as errors. When they are discovered in a given dataset, they might become a foundation for a deeper exploration as far as they might contain some important (yet to be discovered) knowledge. Unfortunately, if such type of data is not removed from the dataset, it has a negative influence on other (further) processes of data analysis. Outliers are capable of decreasing the quality of knowledge mined from a given dataset. Especially, if we rely on the knowledge mined from the clusters, inducted from the data with outliers inside. Of course it also depends on the clustering algorithm used to group the original data. However, there are some algorithms (e.g. *k*-means) which are suspectible to occurrence of outliers. The author works with hierarchical algo-

[1] An operator & is equivalent to logical and.

rithms which are, fortunately, resistant to the outliers. Simply, when there are some unusual data, dissimilar to all the other in the dataset, they are clustered at the end of the clustering procedure. Thus it is enough to set a specified stop criterion (i.e. the moment in which the similarity between the merged clusters is smaller than a given threshold value) which will break the clustering at an exact time. Then, all the nodes (clusters) in a created hierarchical structure are being treated as outliers. The article presents the analysis of the influence of different clustering parameters on the results of the final structure of clusters and their ability to mine the outliers in KBs.

1.1. Outliers in a Knowledge Base

There are numerous papers on mining outliers in data but there seem to be no publications which cover the issue of finding outliers in a specific kind of data such as rules usually stored in KBs. When we deal with outliers in rules we should think about rules which represent some crucial but rare, part of domain knowledge. As a matter of fact, a rule can be defined as a formula with two parts: conditional (premises) and decisional (conclusion). In this context an outlier can be a rule which is dissimilar to all other rules in a given KB, because the set of premises and/or conclusion is dissimilar to the rest of the data (rules). Additionally, outliers can be described as all the rules which contain a much smaller or greater number of premisses when compared with the others. We may also say that an outlier-based rule is one, which consists of unusual attributes and values of such attributes in the conditional part of it as well as in the decisional one. It means that the both types of rules: non-deterministic and the ones with different premises can be treated as outliers. An outlier is not only a single object as, in this case, a single rule in a KB. When we group rules together, based on the similarity criteria (as it is done when using data mining algorithms like cluster analysis), as a result we get a small group (quite often a singleton) while other (more similar) rules create bigger groups. Therefore, during the analysis it is possible that such a small group will be not taken into account and this is not a desirable solution. If we cluster rules in order to optimize the exploration process, outlier-type rules, in a given KB, may take the form of small clusters or a single rule which has not been merged with the others.

2. Knowledge base and rule-based knowledge representation

Rules in a given KB can be given apriori by a domain expert or generated automatically from a dataset using a dedicated algorithm. In this research, we assume that the rules are achieved through the execution of one of many possible algorithms. A very brief introduction to this subject is presented in this Section. Such a natural way of knowledge representation makes rules easily understood by experts and knowledge engineers as well as people not involved in the expert system building. The knowledge represented by the rules should be cohesive and, if possible, should describe all possible cases that can be met during the inference process.

2.1. Inducing the rules from the original data

There are many existing algorithms for generating rules (so called *rule induction* methods) like $LEM1$, $LEM2$, and AQ [3]. Usually, the original datasets have a form of a so-called decision table. $LEM2$ (Learning from Examples Module version 2) algorithm is an example of a global rule induction algorithm and the option of $RSES$ (Rough Set Exploration System) software [4]. It is most frequently used since (in most cases) it produces better results. In general, $LEM2$ computes a local covering and then converts it into a rule set. It explores the search space of attribute-value pairs, its input data file is a lower or upper approximation of a concept (for definitions of lower and upper approximations of a concept see, [3]), so its input data file is always consistent. Original data, stored, for example, in the form of decision table, is used to generate rules from it. By running the $LEM2$ algorithm for such data the decision rules are obtained.

As an example let us take a dataset used for contact lenses fitting, which contains 24 instances, described by 4 nominal attributes: *age of the patient*: (1) young, (2) pre-presbyopic, (3) presbyopic, *spectacle prescription*: (1) myope, (2) hypermetrope, *astigmatic*: (1) no, (2) yes and *tear production rate*: (1) reduced, (2) normal and a decision attribute with 3 classes 1 : *hard contact lenses*, 2: *soft contact lenses* and 3:*no contact lenses*. Class distribution is following: 1: 4, 2: 5 and 3: 15. The piece of the original dataset is as follows:

```
1  1  1  1  1  3
2  1  1  1  2  2
```

```
...
24 3  2  2  2  3
```

Using the $RSES$ system with the $LEM2$ algorithm's implementation the KB with 5 rules has been achieved. The source file of the KB is as follows:

```
RULE_SET lenses
ATTRIBUTES 5
 age symbolic
...
 contact-lenses symbolic
DECISION_VALUES 3
none
soft
hard
RULES 5
(tear-prod-rate=reduced)=>(contact-lenses=none[12]) 12
(astigmatism=no)&(tear-prod-rate=normal)&(spectacle-prescrip=
...
(spectacle-prescrip=myope)&(astigmatism=no)
&(tear-prod-rate=normal)&(age=young)
=>(contact-lenses=soft[1]) 1
```

The rule: `(tear-prod-rate=reduced)=>(contact-lenses=none[12]) 12` should be read as: ***if*** *(tear-prod-rate=reduced)* ***then*** *(contact-lenses=none)* which is covered by 12 of instances in the original dataset (50% of instances cover this rule).

2.2. Managing the rules

The domain experts need to have a proper tool to manage the rules effectively. In other words, there should not be such a situation, in which, for some input knowledge there are no rules to be activated, as it would mean that there is no new knowledge explored from this particular KB for given input data. They (domain experts) need to be able to easily obtain the information about all uncharted areas in an explored domain, in order to complete it as soon as possible. Many papers show the results of clustering of a large set of data but rather rarely for such a specific type of data like a rule-based knowledge representation. From a knowledge engineer's point of view, it is important to come up with a tool which helps to manage the consistency and completeness of the created KB. Decision support systems (DSS), which are usually based on rule-based KBs, use rules to extract new knowledge - this process is called the inference process. Instead of searching every rule one by one, it is possible to find the most relevant group of rules (the representative of such a group of rules matches the given information in the best possible way) and reduce the time necessary to give an answer to a user of such a system. The results obtained in the authors's previous research [5] show that in an optimal case, it was possible to find a given rule when only a few percent of the whole KB has been searched. As long as so much depends on the quality of representatives for groups of rules, it is necessary to choose the best possible clustering algorithm - the one which creates optimal descriptions for groups of rules.

2.3. The Structure of the Article

The rest of the paper is organized as follows. In Section 3 the definition and the assumptions about discovering outliers in KBs are presented. Section 4 includes the pseudocode of the rules clustering algorithm algorithm as well as the description of various similarity and clustering methods which then are examined and presented with the results in Section 6. Section 5.1 introduces the outlier detection method for a hierarchical structure of rules in KBs.

3. Outliers: the Definition, the Meaning and the Types of Outliers

The issue of outlier detection has numerous important uses in many applications which are high-dimensional domains and the data therein may consist of hundreds of dimensions. In paper [6] the authors present a new approach to the summarization of databases containing both numerical and partly standardized textual records. The described method enables the detection of outlier information using linguistic summaries. It is a primary step in many data mining applications. Although outliers are often considered as an error or noise, they may carry important information. Hawkins defines an outlier as an observation that deviates so much from other observations as to arouse suspicion that it has been generated by a different mechanism [7]. Other researchers indicate that an outlying observation, is one that appears to deviate markedly from other members of the sample in which it occurs or an observation in a dataset which appears to be inconsistent with the remainder of that set of data. In case of complex data using statistical methods, based on regression analysis is impossible. That is why we need to find methods which deal with the complexity of the analyzed data, in order to identify all possible outliers.

3.1. The Meaning of Outliers

Outliers may contain important information, therefore they should be investigated carefully. It is well known that quite often they contain valuable information about the process being investigated or the data gathering and recording process. Before considering possible elimination of these points from the data, one should try to understand why they have appeared in the first place and whether it is likely that similar values will continue to appear. Of course, outliers are often bad data points and such as they should be removed from the entire dataset. Outliers can actually represent unexpected factors of practical importance and can therefore contain valuable information. In these situations, the influence of outliers should be emphasized rather than limited or minimized. Any attempts to reduce the influence of outliers without due consideration in these applications can lead to a loss of information which may often be crucial for problem solving.

3.2. Types of Outlier

Outliers can be presented with *scores* as well as with a *label*. Outliers with scores give us the information about a degree of outlierness of each data. Labelling the outliers means that we include the information whether the data is anomalous or not.

A *label* is usually a binary output, indicating whether or not a data point is an outlier. Thus, scores are more informative to analysts and they can be readily converted into labels by choosing a particular threshold. It is easy then, to convert scores to labels.

A *score* quantifies the tendency for a data point to be considered an outlier. Higher values of the score make it more likely that a given data point is an outlier. Some algorithms may even output a probability value quantifying the likelihood that a given data point is an outlier. The form of outlier score can be either the distance of a data point to its nearest neighbor, or local density of a data point or the probabilistic fit values are used to quantify the outliers scores of data points [8, 9, 10].

If we have a continuous data one of the simplest way to determine the outlier score, and/or label is using the Tukey's fences. It requires to calculate the interquartile range ($IQR = Q_3 - Q_1$) which is usually used as a measure of how spread-out the values are. If we assume that the values in a given dataset are clustered around some central value, then the IQR tells how spread out the „middle" values are and it allows to discover the outliers, as data lying outside the range. That is, if a data point is below $Q_1 - 1.5 * IQR$ or above $Q_3 + 1.5 * IQR$, it is viewed as being too far from the central values to be reasonable. Then it will be easy to label data with „normal data" or „outlier" names [11].

3.3. Methods for outlier detection

Most of the approaches to anomaly detection in data mining utilize the distance and similarity to the nearest neighbors and label observations as outliers or non-outliers[7]. There are many different approaches to detect the outliers in data. One of them, analyzed and used by the author of this research is clustering-based method. Clustering refers to unsupervised learning algorithms which do not require pre-labeled data to extract rules for grouping similar data instances [12]. Outlier mining could be defined as the process of grouping sets of records that behave in a different or deviant manner in comparison to the rest or majority of the data. It could also be viewed as the process of clustering, but with the difference that here clusters look out for objects or records that show a different behavior when compared to the rest of the data. There are following important assumptions made whenever we use clustering to detect anomalies:

- When clusters of normal data are being created, any new data that do not fit well with existing clusters of normal data are considered anomalies.

- Even if created clusters contain both normal and anomalous data, we still are able to recognize the outliers. The normal data lie close to the nearest clusters centroid whereas anomalies are far away from centroids.

- Created clusters have various sizes, thus setting some threshold for a size of created clusters we are able to decide whether a given cluster is outlier if its size or density is below a threshold.

Different clustering algorithms are proposed in the research related to the outlier detection as a result of the clustering process. In [13] the approach which uses k-means clustering to generate normal and anomalous clusters i presented. Then, the created clusters are analyzed further to discover outliers. An instance is classified as normal, if it is closer to a normal cluster's centroid and vice versa (if the distance between an instance and centroid is larger than a predefined threshold, the instance is treated as an anomaly). When clustering is achieved, assuming that normal instances constitute a larger proportion of the entire dataset, a given percent of clusters are normal and the rest are anomalous. Other approaches, based for example on the density- and grid-based clustering algorithm are proposed in [14]. It is worth to go through the research presented in [15]. The authors have investigated the performances of various clustering algorithms (five different approaches: the *k*-means, improved *k*-means, *k*-medoids, Expectation Maximization (*EM*) clustering, and distance-based anomaly detection algorithms) when applied to anomaly detection.

As outlier mining seems to be so easy to manage using the cluster analysis method it is necessary to make some notes about this kind of data mining techniques in this paper. Thus, in the next section (Section 4), the general idea, the pseudocode and the most important aspects of clustering algorithms are given. Then, the pseudocode of the outlier mining algorithm is given in Section 5.1.

4. Clustering Algorithms

A cluster is a collection of objects which are similar to one another and dissimilar to the objects belonging to other clusters. Moreover, a clustering algorithm aims to find a natural structure or relationship in an unlabeled data set. There are several categories of clustering algorithms. Some of the algorithms are hierarchical and probabilistic. In this paper the author presents a hierarchical clustering algorithm which is based on the connection between the two nearest clusters. The starting condition is carried out by setting every object as a separate cluster. In each step, the two most similar objects are merged, and a new cluster is created with a proper representative for it. After a specified stop condition is reached the clustering process for the rules (or their groups) is finished. There are many possible ways for defining the stop condition. For instance it can be the reaching of a specified number of groups, or the moment in which the highest similarity is under a minimal required threshold (which means the groups of rules are now more differential than similar one another).

The pseudocode of the hierarchical clustering algorithm - namely Classic AHC (agglomerative hierarchical clustering) algorithm [16] - is presented as Pseudocode 1.

Pseudocode 1. Classic AHC Algorithm.
Input: stop condition *sc*, ungrouped set of objects *s*
Output: grouped tree-like structure of objects

```
1. Place each object $o$ from $s$ into a separate cluster.
2. Build a similarity matrix $M$ that consists of similarity values for each pair of clusters.
3. Using $M$ find the most similar pair of clusters and merge them into one.
4. Update $M$.
5. IF $sc$ is met end the procedure.
6. ELSE REPEAT from step 3.
7. RETURN the resultant structure.
```

The most important step is the second one, in which the similarity matrix *M* is created on the basis of the selected similarity measure and a pair of the two most similar rules (or groups of rules) are merged. In step 1 two parameters are given by a user: the similarity measure and the clustering method. Eventually, both of them result in achieving different clusterings. For this reason the author decided to compare similarity measures in this research. In order to do that, the author choose five different similarity measures and repeated the clustering algorithm many times for each of them while changing the number of groups [2] as well as the clustering method.

The main advantage of hierarchical clustering is that it does not impose any special methods of describing the clusters similarity.

[2] In this work clustering is stopped when a given number of clusters is generated.

4.1. Rules Clustering Algorithms

Clustering algorithms allow to organize the rules in a smart way [16]. To achieve groups of similar rules it is necessary to propose a method of deciding which rules are the most similar in a given step of the clustering process. Because rules are a specific type of data, usually attributed with short descriptions, the differences between rules are difficult to notice. Hence it is so important to find a similarity measure which is able to find all the differences and, as a result, decide about the order of rules clustering in an optimal way.

4.2. Similarity Analysis

In the literature there are numerous methods of describing similarity between objects [17] that can be modified to work with rules as well. The similarity measure used to find a pair of rules or groups of rules that are the most similar in a given moment is called the *intra-cluster similarity measure*. The author studied the following five measures: Simple Matching Coefficient (SMC), the Jaccard Index (based on the SMC measure) sometimes also called the weighted similarity or the weighted similarity coefficient (denoted herein as $wSMC$ [18], the *Gower* measure (widely known in the literature) [19] and two measures taken from the retrieval information domain: inverse occurrence frequency (IOF) and occurrence frequency (OF). It is crucial to answer the question if a given similarity measure influences the shape of a grouped KB's structure. Measuring a similarity or a 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 aggregate both types at once the problem is much more complicated.

For a set of attributes A and their values V, rules premises and conclusions are built using pairs (a_i, v_i). In this approach $a_i \in A, v_i \in V_a$ and a pair (a_i, v_i) is called a descriptor. In a vector of such pairs, i-th position denotes the value of the i-th attribute of a rule. Most of the rules do not consist of all attributes in A, thus constructed vectors (describing the rules) are of different lengths.

Almost all similarity measures assign a similarity value between two rules r_j and r_k belonging to the set of rules R as follows:

$$S(r_j, r_k) = \sum_{i=1}^{n} w_i s(r_{ji}, r_{ki}) \tag{1}$$

where $s(r_{ji}, r_{ki})$ is the per-attribute (for i-th attribute) similarity between two values of descriptors of the rules r_j and r_k, and n is the number of attributes analyzed when the similarity is calculated. Of course the range for j and k is given as the number of rules in the knowledge base. The quantity w_i denotes the weight assigned to the attribute a_i and usually $w_i = \frac{1}{d}$, for $i = 1, \ldots, d$. In all the definitions, the s_{jki} denotes the contribution provided by the i-th variable.

The simplest measure is SMC (Simple Matching Coefficient) [3] - which calculates the number of attributes that match in the two rules in the following way:

$$s_{SMC}(r_{ji}, r_{ki}) = s_{jki} = 1 \texttt{ if } r_{ji} = r_{ki} \texttt{ else } 0. \tag{2}$$

The range of the per-attribute SMC is $\{0; 1\}$. It deals with all types of attributes in the same way. Unfortunately it tends to favor longer rules thus it is better to use the $wSMC$ measure, which is similar to SMC. However it is more advanced as it also divides the result by the number of attributes of both objects so longer rules are not favored any more. It can be defined in the following form:

$$s_{wSMC}(r_{ji}, r_{ki}) = s_{jki} = \frac{1}{n} \texttt{ if } r_{ji} = r_{ki} \texttt{ else } 0 \tag{3}$$

where n is the number of attributes. The *Gower* similarity coefficient is the most complex of the all used inter-cluster similarity measures as it handles numeric attributes and symbolic attributes differently. For ordinal and continuous variables it defines the value of s_{jki} as $s_{jki} = 1 - \frac{|r_{ji} - r_{ki}|}{range(i)}$, where: $range(i)$ is the range of values for the i-th variable. For continuous variables s_{jki} ranges between 1, for identical values $r_{ji} = r_{ki}$ and 0 for the two extreme values r_{max} - r_{min}.

Deriving from *retrieval information systems*, two measures could be also used to check the similarity between rules (or clusters of rules): IOF and OF. The first one (IOF) assigns a lower similarity to mismatches on more frequent values while the second one (OF) gives opposite weighting for mismatches when compared to the IOF measure, i.e. mismatches on less frequent values are assigned a lower

[3] If both compared objects have the same attribute and this attribute has the same value for both objects then add 1 to a given similarity measure. If otherwise, do nothing. To eliminate one of the problems of SMC, which favors the longest rules, the author has also used the $wSMC$ measure.

similarity and mismatches on more frequent values are assigned a higher similarity. They can be defined as follows:

$$s_{IOF}(r_{ji}, r_{ki}) = \begin{cases} 1 & \text{if } r_{ji} = r_{ki}; \\ \frac{1}{1+\log(f(r_{ji}))\cdot\log(f(r_{ki}))} & \text{if } r_{ji} \neq r_{ki}. \end{cases}$$

and

$$s_{OF}(r_{ji}, r_{ki}) = \begin{cases} 1 & \text{if } r_{ji} = r_{ki}; \\ \frac{1}{1+\log\frac{n}{f(r_{ji})}\cdot\log\frac{n}{f(r_{ki})}} & \text{if } r_{ji} \neq r_{ki}. \end{cases}$$

4.3. Clustering methods

The distance among clusters can be computed using different methods, among which the following four methods are the most popular: *Single Linkage* (SL), *Complete Linkage* (CoL), *Average Linkage* (AL) and *Centroid-based Linkage* (CL)[16]. SL is a method that focuses on the minimum distances or the nearest neighbor between clusters meanwhile CoL concentrates on the maximum distance or the furthest neighbor between clusters. AL is a compromise between the sensitivity of CoL to outliers and the tendency of SL to form long chains that do not correspond to the intuitive notion of clusters as compact, spherical objects. In the CL method, the centroid is the mean of all points in a cluster.

4.4. Cluster Validity

Cluster validity seems to be an important feature in checking whether the created structure has got a good quality. If not, it may treat some data as outlier even it is not one. There are two criteria which are necessary to meet when good clustering results have to be achieved: separation and cohesion. There are many different measures to check if both of these conditions are met. One of the most popular is the MDI index, which has been examined in this research but unfortunately did not bring any valueable information. Thus the author decides to use other cluster validity indexes in future research to determine a best clustering structure for a given set of data.

5. Outlier Mining Using Clustering Algorithms

An outlier or a noise point is an observation which appears to be inconsistent with the remainder of the data. Outliers may be considered as noise points lying outside a set of defined clusters. Generally, existing techniques work well in the absence of noise. When there is noise in the dataset, the clusters identified by these techniques include the surrounding noise points too. The presence of noise disrupts the process of clustering. Noise has to be separeted from the dataset to enhance the quality of the clustering results.

5.1. Outlier Detection Algorithm

The pseudocode of the outlier detection algorithm based on clustering approach is presented as Pseudocode 2.

Pseudocode 2. Outlier detection algorithm.
Input: M - Number of clusters, $G = \{g_1, g_2, \ldots, g_M\}$.
Output: *UngroupedCount* - a list of ungrouped rules.

```
1. Group data objects $X$ using the $AHC$ algorithm in order to achieve $M$ clusters grouped in tree-like structure, $UngroupedCount = 0$.
2. For each cluster $g_i$ from the $G$ set
3.     IF $sizeOf(g_i) == 1$ THEN $++UngroupedCount$
4. RETURN $UngroupedCount$.
```

The goal of the algorithm is to discover ungrouped rules - individual rules which could not be joined with the others, because they do not have any common feature (neither premises nor conclusion). Only such rules (individual objects) represent something interesting to be found in the KB as it means that there are unique (so-far unexamined) areas of domain knowledge which are totally different from the rest of knowledge already stored in our KB. The most important step is the examining of each of the created cluster g_i in order to check if its size is equal to 1 as it would mean that it is an outlier - an ungrouped rule. The results of this research are given in Section 6.

6. Experiments

The goal of the experiments is to check whether choosing similarity and/or clustering methods influences the possibility of finding outlier-type rules in *KBs*.

Thus, in this section, experimental evaluation of 5 similarity measures and 4 clustering methods (described in Section 4) on 7 different *KB*s [20] is presented. Decision rules have been generated from the original data using RSES software and *LEM*2 algorithm [4]. The smallest number of attributes was 5, while the greatest was 280. The smallest and the greatest number of rules were 42 and 490 respectively. The details of the analyzed datasets are included in Table 1. The acronyms for the datasets are as follows: arythmia (A), audiology (B), autos (C), balance (D), Breast cancer (E), diab (F), diabetes (G).

Table 1: Data gathered in the experiments.

	#C	*#A*	*#R*	*#N*	BCS	*#U*
A	12.5 ± 2.5	280	154	295.5 ± 2.5	111.5 ± 42.0	5.8 ± 4.7
B	6.9 ± 3.0	70	42	77.2 ± 3.0	30.0 ± 7.9	3.6 ± 2.7
C	7.8 ± 2.4	26	60	112.2 ± 2.4	37.9 ± 14.3	3.8 ± 3.0
D	19 ± 9.1	5	290	560 ± 9.1	170 ± 95	6.9 ± 9.0
E	11 ± 1.0	10	130	240 ± 1.0	76 ± 32	5.1 ± 3.4
F	29 ± 19	9	480	940 ± 19	320 ± 130	11 ± 13
G	29.5 ± 19.7	9	490	950.5 ± 19.7	336.6 ± 135.2	12.5 ± 14.4

The meaning of the columns in Tables 1, 2 and 3 is as follows:

- *#A*- number of different attributes occurring in premises or conclusions of rules in a given knowledge base.
- *#R* - number of rules in an examined knowledge base.
- *#C* - number of clusters created during the clustering algorithm.
- *#U* - number of singular clusters in the resultant structure of grouping.
- *#N* - number of nodes in the created structure.
- *BCS* - Biggest cluster's size - number of rules to have been used in the cluster.

The first experiment is based on comparison of five similarity measures: SMC, $wSMC$, $Gower$ and two measures well known in the area of retrieval information systems: IOF and OF. The goal of the experiment was to check if a given similarity measure influences the possibility of finding outliers in rules (the results are included in Table 2). The same analysis has been carried out for the clustering methods (see the results in Table 3). To reach that goal, apart from calculating the number of outliers (ungrouped rules in clustering process), the percentage of outliers in the number of rules has been measured in the fallowing way:

$$Outlier\_vs\_Rules = \frac{\#U}{\#R}.$$

Then three cases were analyzed:

- >= 1% - when the percentage of outliers in the whole *KB* was at least equal to 1%.
- >= 5% - when the percentage of outliers in the whole *KB* was at least equal to 5%.
- >= 10% - when the percentage of outliers in the whole *KB* was at least equal to 10%.

Tables 2 and 3 present the frequency of each analyzed similarity measure and clustering method in regard to being able to find a given number (percentage) of outliers.

Table 2: Outlier detection vs. similarity measures.

	>= 1%	>= 5%	>= 10%
Gower	41(21.47%)	24(21.82%)	50(19.61%)
IOF	41(21.47%)	24(21.82%)	52(20.39%)
OF	41(21.47%)	25(22.73%)	52(20.39%)
SMC	33(17.28%)	19(17.27%)	50(19.61%)
wSMC	35(18.32%)	18(16.36%)	51(20%)
Total	191(68.21%)	110(39.29%)	255(91.07%)

In Table 2, it is easy to notice the tendency of two similarity measures (SMC and its modification $wSMC$) to discover the smallest number of outliers if compared to other measures. As Table2 shows, when the percentage of outliers is at least equal to 5%, it is predominantly by using *Gower* or *IOF* and *OF* similarity measures while the frequency of the remaining measures was lower. It seems that if we do not want to discover too many outliers after the outlier mining process, these two measures should be used.

The comparison of the four clustering methods: SL, CL, CoL and AL and its influence on the frequency of discovering the outliers is presented in Table 3.

Table 3: Outlier detection vs. clustering methods.

	>= 1%	>= 5%	>= 10%
SL	68(35.6%)	45(40.91%)	61(23.92%)
CL	22(11.52%)	10(9.09%)	70(27.45%)
AL	43(22.51%)	14(12.73%)	65(25.49%)
CoL	58(30.37%)	41(37.27%)	59(23.14%)
Total	191(68.21%)	110(39.29%)	255(91.07%)

Table 3 shows the tendency of generating the smallest number of outliers when the CL or AL clustering method is used (for cases with 1% and 5% of outliers). The clusterings with many outliers discovered, are most often those, when the SL method is used - which is well described in the literature. Surprisingly this situation has been observed only when the outliers fill ≥ 1% and ≥ 5% of the *KB*'s size. When the number of outliers is at least equal to 10% - it is most often achieved for the CL and AL methods. It requires further research, which will be the author's future subject of research.

The second experiment has been based on the analysis of values of such parameters as: biggest cluster size (BCS), the number of outliers (#*U*) and above described, the percentage of outliers in rules (*Out_vs_R*). The results of this analysis are presented in Tables 4 and 5 separately for similarity and clustering methods.

Table 4: Outlier detection vs.clustering methods.

	BCS	#*U*	*Out_vs_R*
SL	212.89 ± 166.76	11.90 ± 11.64	0.06 ± 0.04
CL	98.29 ± 107.84	2.19 ± 3.17	0.02 ± 0.03
AL	152.21 ± 137.41	4.14 ± 4.57	0.03 ± 0.04
CoL	155.49 ± 137.20	9.40 ± 10.08	0.06 ± 0.05

Table 4 confirms the lack of any significant differences between using the analyzed similarity measures in the context of their influence on the number of clusters, the number of ungrouped rules (outliers) and other parameters. The only conclusion we can come up with is the following: using SMC and $wSMC$ measures gives similar results, while using *Gower* measure gives us a higher number of outliers ($\#U$), and a slightly greater biggest cluster's size (BCS).

Table 5: Outlier detection vs. similarity measures.

	BCS	$\#U$	Out_vs_R
Gower	157.23 ± 147.43	8.05 ± 9.92	0.05 ± 0.05
IOF	157.89 ± 145.83	7.96 ± 10.07	0.05 ± 0.05
OF	157.68 ± 146.59	7.73 ± 10.08	0.04 ± 0.04
SMC	155.63 ± 143.62	5.57 ± 6.72	0.04 ± 0.04
wSMC	145.16 ± 141.09	5.21 ± 7.71	0.04 ± 0.05
Total	154.72 ± 143.97	6.91 ± 9.03	0.04 ± 0.04

Table 5 confirms the existance of significant differences between using the analyzed clustering methods (SL, CL, CoL, AL in the context of their influence on the examined parameters: $\#U$, BCS and Out_vs_R. At the significance level of $p < 0,05$, it is possible to conclude that the SL method brings the biggest BCS and $\#U$ in comparison to the other clustering methods, while the CoL method produced the smallest. The results confirm all the features of these methods which can be widely found in the literature. Thus to get higher number of outliers we should choose SL instead of the other methods.

Figures 1 and 2 confirm all the remarks made in the results of the experiments.

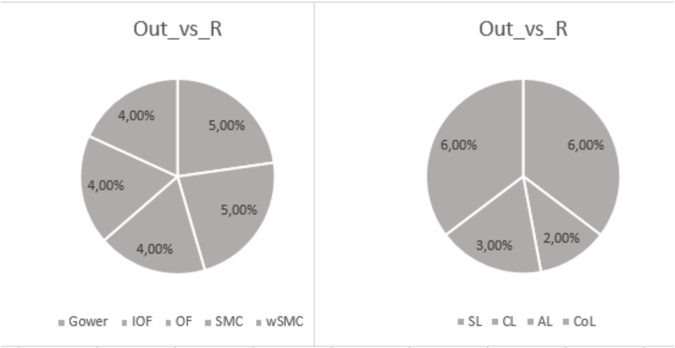

Figure 1: The percentage of outliers vs. similarity measures and clustering methods.

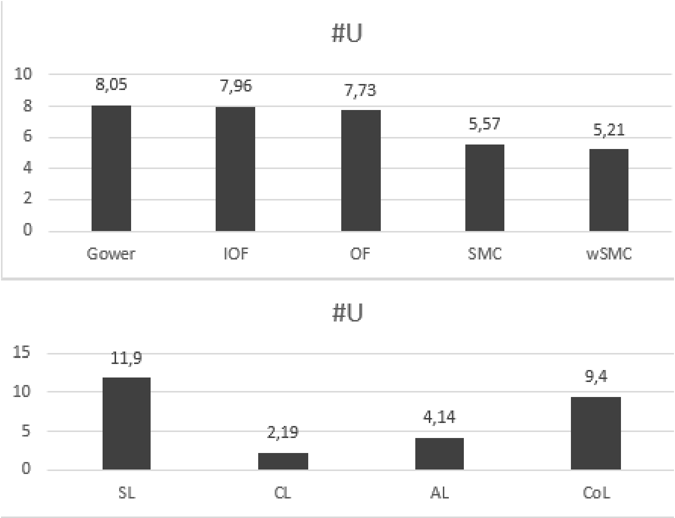

Figure 2: Average number of outliers vs. similarity measures and clustering methods.

7. Summary

This article presents the evaluation of five different similarity measures and four clustering methods used for comparing the results of clustering of rules in KBs. The experiments have been carried out for seven different KBs from different domains and such datasets tend to differ in many parameters. Rules have been clustered using the AHC algorithm presented in Section 4. The results taken from the experiments have been compared by the following parameters: the number of clusters, the number of ungrouped rules, the size of the biggest cluster and the percentage of outliers in the size of the KB. The description of the analyzed data is included in Table 1 whereas the results of the experiments are presented in Tables 2, 3, 4, 5 and in Figures 1 and 2. The selection of similarity and clustering method to be used is crucial and there is a strong correlation between using a given clustering parameter and the value of the following parameters: the biggest clusters size, the number of outliers in rules as well as the percentage of outliers in the number of rules in a given KB. As it can be found in the literature, the SL results in achieving a higher number of outliers, while the CoL has a tendency to give the smallest number of outliers. In future research the author plans to examine other, much varied, similarity measures, and check if the size of the input data: the number of rules, the length of the rules, the type of the attribute used to describe rules in *KBs* has got any influence on the efficiency of the outlier mining process.

References

[1] Portnoy, L., Eskin, E., and Stolfo, S., *Intrusion detection with unlabeled data using clustering*, In: In Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001, 2001, pp. 5–8.

[2] Pedrycz, W., *Knowledge-based clustering - from data to information granules*, Wiley, 2005.

[3] Grzymala-Busse, J. W., *A New Version of the Rule Induction System LERS*, Fundam. Inf., Vol. 31, No. 1, July 1997, pp. 27–39.

[4] Bazan, J. G., Szczuka, M. S., and Wroblewski, J., *A New Version of Rough Set Exploration System*, In: Rough Sets and Current Trends in Computing, Third International Conference, RSCTC 2002, Malvern, PA, USA, October 14-16, 2002, Proceedings, 2002, pp. 397–404.

[5] Nowak-Brzezińska, A., *Mining Rule-based Knowledge Bases Inspired by Rough Set Theory*, Fundamenta Informaticae, Vol. 148, No. 35, 2016, pp. 35–50.

[6] Duraj, A., Szczepaniak, P., and Ochelska-Mierzejewska, J., *Detection of Outlier Information Using Linguistic Summarization*, In: Flexible Query Answering Systems 2015; Advances in Intelligent Systems and Computing 400,(Eds.: Andreasen T., et al.), Proceedings of the 11th International Conference FQAS 2015, 2016, pp. 101–113.

[7] Hawkins, D., *Identification of Outliers*, Chapman and Hall, 1980.

[8] Aggarwal, C. C. and Sathe, S., *Outlier Ensembles - An Introduction*, Springer, 2017.

[9] Aggarwal, C. C., *Data Mining - The Textbook*, Springer, 2015.

[10] Aggarwal, C. C., *Outlier Analysis*, Springer, 2013.

[11] Tukey, J. W., *Exploratory Data Analysis*, Addison-Wesley, 1977.

[12] Jain, A. K., Murty, M., and Flynn, P., *Data clustering: A review*, ACM Computing Surveys, Vol. 31, No. 3, 1999, pp. 264–323.

[13] Münz, G., Li, S., and Carle, G., *Traffic Anomaly Detection Using KMeans Clustering*, In: In GI/ITG Workshop MMBnet, 2007.

[14] Leung, K. and Leckie, C., *Unsupervised Anomaly Detection in Network Intrusion Detection Using Clusters*, In: Proceedings of the Twenty-eighth Australasian Conference on Computer Science - Volume 38, ACSC '05, Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 2005, pp. 333–342.

[15] Syarif, I., Prugel-Bennett, A., and Wills, G., *Unsupervised Clustering Approach for Network Anomaly Detection*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 135–145.

[16] Jain, A. K. and Law, M. H. C., *Data Clustering: A User's Dilemma*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 1–10.

[17] Boriah, S., Chandola, V., and Kumar, V., *Similarity measures for categorical data: A comparative evaluation*, In: In Proceedings of the eighth SIAM International Conference on Data Mining, pp. 243–254.

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

[19] Gower, J. C. and Gower, J. C., *A general coefficient of similarity and some of its properties*, Biometrics, 1971.

[20] Asuncion, A. and Newman, D., *UCI Machine Learning Repository*, 2007.