Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 176 (2020) 1420–1429

Procedia Computer Science

www.elsevier.com/locate/procedia

24th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Outliers in rules - the comparision of LOF, COF and KMEANS algorithms.

Agnieszka Nowak - Brzezińska[a,*], Czesław Horyń[a]

[a] *University at Silesia, Katowice, Bankowa 12, Poland*

Abstract

The aim of the article is the analysis of using *LOF*, *COF* and *Kmeans* algorithms for outlier detection in rule based knowledge bases. The subject of outlier mining is very important nowadays. Outliers in rules mean unusual rules which are rare in comparison to others and should be explored further by the domain expert. In the research the authors use the outlier detection methods to find a given (1%, 5%, 10%) number of outliers in rules. Then, they analyze which of seven various quality indices, that they used for all rules and after removing selected outliers, improve the quality of rule clusters. In the experimental stage the authors used six different knowledge bases. The results show that the optimal results were achieved for *COF* outlier detection algorithm as the one for which, among all analyzed quality indices, the cluster quality improved most frequently.

Keywords: outliers, LOF, COF, quality indices, clustering ;

1. Introduction

Our scientific interests relate to rule knowledge bases and processes of extracting new knowledge from such bases. In case of large knowledge bases, the efficiency of the inference process requires appropriate reorganization of the rules, e.g. into a structure of clusters of rules similar to each other. We assume that after clustering the rules, each cluster will be headed by a representative. Then in the inference process we will only review representatives of rule clusters, which will significantly reduce the time of inference. The efficiency of inference depends not only on the number of rules (affecting the time of inference) but also on the quality of the clusters of rules and their representatives. We dealt with these studies, among others at work [9] and [10]. These works do not, however, include outlier detection algorithms in rule-based knowledge bases. The quality of clustered rules depends on the consistency (similarity) of the grouped rules. When we cluster the rules which are not similar enough, the quality of rule clusters and their representatives is deteriorated, which results in decreasing the efficiency of inference. For this reason, we

[*] Corresponding author. Tel.: +48-32-3689757 ; fax: +48-32-3689866.
E-mail address: agnieszka.nowak@us.edu.pl

10.1016/j.procs.2020.09.152

became interested in the impact of unusual rules on the quality of clusters. In order to review this we analyzed many different outlier mining algorithms and finally we chose 3 methods: LOF, COF, KMeans. They were previously used for typical simple data. Then we used them to examine the effectiveness of detection of unusual (outlier) rules as well as the sensitivity of selected measures of cluster quality to the presence of outliers in the data. Such rules have to be discovered by knowledge engineers and discussed with knowledge experts. Knowledge experts knowing the rules which are unusual in some context can extend the domain to which an unusual rule belongs. In other words, the fact that we are anable to discover the rules which are unusual in some context brings the opportunity to extend the domain knowledge. For domains such as medicine, economy etc. is a very important issue.

The structure of the article is following. In Section 2 we present the research objectives not only related to this work but also to subsequent ones. Rule clustering is the subject of Section 3. Section 4 introduces outlier detection algorithms while Section 5 describes the experimental part of our research with the analysis of the experiments results.

2. Rule-based knowledge bases and inference processes

In our research we analyze knowledge bases with rule representation and we try to develop the methods for efficient exploration of such rules. Rules in the form $IF - THEN$ are very popular methods for representation of experts' knowledge. No matter who uses the knowledge, physicians or economists, rules can be convinient method for various domains. Rules are probably the most natural way of explaining the way of domain experts' thinking. Rules can create chains in such a way that the conclusion of one rule can then be a premise in another one. When the rules form such chains in the knowledge base, their analysis can be complicated. The real nature of this type of data means that unusual need not to be false rules. It is enough that they relate to a rarely explored area of domain knowledge and will be significantly different from the rest of the rules. Finding them in the set of all rules can allow a knowledge engineer to take care of the completeness of the knowledge base and succesfully implemented inference process. What is more, most often rules (their premises and conclusions) are created using various types of attributes: quantitative, qualitative, binary. In addition, some rules may be very short (containing several premises) while others may contain multiple premises. All this makes the rules very unusual data structures and require specialized analysis. An efficient exploration of rules requires reorganization of them in the structure of rule clusters. In order to do that we need to use a proper clustering algorithm. After analysing various clustering method we finally choose the hierarchical method with agglomerative type (described in Section 3) because because it intuitively corresponds to the idea of clustering similar rules into groups.

3. Rule clustering

The arsenal of hierarchical clustering is extremely rich. The process of hierarchical clustering can follow two basic strategies: agglomerative and divisive. Since the agglomerative technique is more natural when we want to cluster the data until they are similar enough, further in this research we focus only on this method. The agglomerative algorithms consider each object as a separate cluster at the outset, and these clusters are fused into larger and larger clusters during the analysis, based on between-cluster or other (e.g., homogeneity) measures. In the last step, all objects are amalgamated into a single, trivial cluster. In each algorithmic step, the nearest pairs of objects or clusters are found and then amalgamated. After the fusion, the distances between the newly obtained clusters and all the other clusters and objects are recalculated [1]. The crucial part of the analysis is the way these new distances are calculated. To complete this calculation, the recurrence formula proposed by Lance - Williams may be used. There are the following methods to use in this case: *single linkage* (The two closest points from each cluster), *complete linkage* (The two farthest points from each cluster), *average linkage*, *WPGMA - Weighted Pair Group Method with Arithmetic Mean* (The average of the clusters distances is taken, not considering the number of points in that cluster, also called McQuitty), *UPGMA - Unweighted Pair Group Method with Arithmetic Mean* (The average of the clusters distances is taken whilst compensating for the number of points in that cluster), *centroid linkage* (The inter-cluster mid-point) and *Ward's* method (Calculates the increase in the error sum of squares (ESS) after fusing two clusters).

[1] Various distance or similarity measures can be used here. In this research the authors used very well known Gower's measure [5]

Which method we choose affects the results [2]. When the knowledge base contains unusual rules it is dificult to build clusters with high quality. Such rules influence the quality of created clusters and we decided to examine this in details.

3.1. Ouliers in rules - the study

There are several topics related to anomaly (outlier) detection. Noise detection, which is processing the data in order to remove unwanted noise so that the patterns in the data could be better studied. However this differs from outlier detection in the sense that in outlier detection we are interested in finding those records rather than filtering them. Novelty Detection, which refers to the detection of new patterns that were previously absent or overlooked. The normal model is usually modified by those patterns which is the major discrepancy between it and outlier detection.There are a number of challenges that makes the outlier detection problem increasingly obscure. In machine learning, the detection of „not-normal" instances within datasets has always been of great interest. This process is commonly known as *anomaly detection* or *outlier detection*. According to the definition given by *Grubbs* in 1969: „An outlying observation, or outlier, is one that appears to deviate markedly from other members of the sample in which it occurs" [11]. Today outliers are known to have two important characteristics: outliers are different from the norm with respect to their features and they are rare in a dataset compared to normal instances. Anomaly detection algorithms are now used in many application domains and often enhance traditional rule-based detection systems. *Intrusion detection* is probably the most well-known application of outlier detection. In this application scenario, network traffic and server applications are monitored. Potential intrusion attempts and exploits should then be identified using outlier detection algorithms. *Fraud detection* is another application of outlier detection. Here, typically log data is analyzed in order to detect misuses of a system or suspicious events indicating fraud. In particular, financial transactions can be analyzed in order to detect fraudulent accounting and credit card payments logs can be used to detect misused or stolen credit cards. With the strong growth in internet payment systems and the increase of offered digital goods, such as ebooks, music, software and movies, fraud detection becomes more and more important in this area. In *medical applications* and life sciences, outlier detection is also utilized. One example is patient monitoring, where electrocardiography (ECG) signals or other body sensors are used to detect critical, possibly life-threatening situations. In life sciences, anomaly detection might also be utilized to find pathologies and mutants [12].

Classic outlier detction algorithms that most often use distance measures for quantitative type of data and also assume all data have the same data representation space are not suitable for rules representation. Real knowledge base has got rules with different structure (different length, and data types). We assume that a given rule is a kind of outlier if its similarity to the others is too small. In the literature there are various similarity measures suitable for rule representation. Correct analysis of the similarity between the rules requires that before starting outlier mining process to transform all rules into a structure in which each rule will be a vector of a fixed length equal to the number of attributes forming the premises. Only indexes corresponding to the attributes that make up a given rule will be analyzed. In the literature on outlier detection algorithms one of he most popular one is LOF algoritm, described in the context of rules in Section 4. Unusual rules are not created in the result of an error but have got an unusual features and in that context differs from other rules in a given knowledge base. Such rules have to be discovered by knowledge engineers and discussed with knowledge experts. Knowledge experts knowing the rules which are unusual in some context can extend the domain to which an unusual rule belongs. In other words, the fact that we are anable to discover the rules which are unusual in some context brings the opportunity to extend the domain knowledge. For domains such as medicine, economy etc. it is very important issue.

4. LOF, COF and KMEANS algorithms

Among many others algorithms for outlier detection we chose the *LOF* (Local Outlier Factor) [13] algorithm as a base for which we propose other algorithms: *COF* [14] and *Kmeans* [15]. The local outlier factor (*LOF*) is based on a concept of the local density, where locality is given by k nearest neighbors, whose distance is used to estimate the density. By comparing the local density of an object to the local densities of its neighbors, one can identify regions of similar density, and points that have a substantially lower density than their neighbors. These are considered to be outliers [13]. Local density based methods compare the local density of the object to that of its neighbors. The definitions of *reachability distance* and *local reachability density* are used in LOF algorithm. The *reachability distance* ($reach-dist(p,o)$) is the maximum of $d(p,o)$ and $k-distance(o)$. The *local reachability density* ((lrd)) of object p relative to $N_k(p)$ is the inverse of the mean reachability distance over the neighborhood set defined as $lrd_{N_{k(p)}}(p) = \frac{|N_k(p)|}{\sum_{o\in N_k(p)} reach-dist(p,o)}$. The *local outlier factor* (LOF) is the ratio between the average local reachability density of the neighborhood to that of the object ((eq. 1)).

$$LOF_{N_{k(p)}}(p) = \frac{\sum_{o\in N_k(p)} lrd_{N_{k(o)}}(o)}{|N_k(p)| lrd_{N_{k(p)}}(p)} \tag{1}$$

The connectivity-based outlier factor (COF) is a local density based approach in order to handle outliers deviating from spherical density patterns [14]. It is a variant of LOF that also uses the k-neighborhood. In contrast to LOF, COF is able to handle outliers deviating from low density patterns. The local density in the COF is the inverse of the average chaining distance. The average chaining distance in contrast to the local reachability distance of the LOF does not use the distance between the point to the points in its neighborhood. During the calculation of the average chaining distance for a point p we distinguish between two sets, the set of objects connected to p and the remainder of the neighborhood set, let P denote the earlier and Q denote the latter. The distance between the two sets and the nearest neighbor of set P in Q are defined as follows. If $o \in P$ and $q \in Q$ then the distance $d(P,Q)$ between the set P and Q is the minimum distance between their elements. q is the nearest neighbor of P in Q ($nearest-neighbor(P,Q)$) if $\exists_o$ such that $d(o,q) = d(P,Q)$. The *average chaining distance* and the COF are computed as $ac-dist_{N_{k(p_1)}}(p_1) = \sum_{i=1}^{r} \frac{2(r-1+1)}{r(r+1)} \cdot CDS_i$, where $r = |N_k(p_1)|$. Detailed information about the cost description sequence(CDS) is included in [16]. The *connectivity based outlier factor* (COF) is the ratio of the *average chaining distance* of p to the mean of the average chaining distance over the $k-neighborhood$ of p (eq. 2).

$$COF_k(p) = \frac{|N_k(p)| ac-dist_{N_{k(p)}}(p)}{\sum_{o\in N_k(p)} ac-dist_{N_{k(o)}}(o)} \tag{2}$$

As observed in the formulas the average chaining distance is the weighted sum of the cost description sequence. Like LOF a score near 1 indicates that the point is not an outlier. A score much greater than 1 indicates outlierness. COF is a variation of LOF, the difference is in calculating the k-distance of the object p, which is calculated in an incremental mode. The COF algorithm has been introduced as a result of observations that, although a high-density set can represent a pattern, not all patterns need to be high-density. Unlike density-based methods, COF can detect exceptions among data of varying density. In addition, the COF algorithm can detect exceptions close to high-density areas. The COF indicates how much the object is isolated from the others and to what extent it can be considered a deviation. In LOF, the nearest neighbors are selected based on Euclidean distance. COF uses the shortest path method, called chain distance. We can also detect outliers using the $k-means$ algorithm [15]. Data is divided into k[2] groups using $k-means$, assigning them to the nearest cluster centers. Then we can calculate the distance between each object and its focus center and choose those with the largest distances as outliers. The clustering algorithm runs iteratively. In the first step, the algorithm randomly selects k-points (objects) as means of clusters. The remaining points are assigned to the cluster for which the Euclidean distance of the object to the center of the cluster is the smallest. In the next step, the means are updated (the average of all clusters) and the step of assigning all objects to the nearest (in the sense of distance) clusters is repeated. The step of updating the means of clusters and assigning objects to the nearest clusters is repeated as long as the objects are relocated to clusters. The algorithm terminates if objects are assigned to the same clusters in subsequent iterations. For each object we select the group to which the object belongs. We calculate the distance between each object and the center of the group, to which belongs. For each group, select objects with the largest distance (in the sense of Euclidean measure) as outliers.

5. Clustering quality

To examine whether we built rule clusters with a good quality we need to measure its quality using one of the following indices. The most popular measures of cluster quality assessment [6] include, among others:

[2] In our case, we take k as the number equal to $DECISION\_VALUES$ (number of decision values) from the loaded knowledge base file.

1. *Dunn index* is an internal evaluation scheme, where the result is based on the clustered data itself. The aim of this index is to identify sets of clusters that are compact, with a small variance between members of the cluster, and well separated, where the means of different clusters are sufficiently far apart, as compared to the within cluster variance. The higher the *Dunn index* value, the better is the clustering. The number of clusters that maximizes the Dunn index is taken as the optimal number of clusters k. It also has some drawbacks. As the number of clusters and dimensionality of the data increase, the computational cost also increases. The Dunn index for c number of clusters is defined as eq. 3:

$$D(u) = min_{1\leq i\leq c}\{min_{1\leq j\leq c, j\neq i}\{\frac{\delta(X_i, X_j)}{max_{1\leq k\leq c}\{\Delta(X_k)\}}\}\} \tag{3}$$

   $\delta(X_i, X_j)$ is the inter-cluster distance between cluster C_i and C_j, and $\Delta(X_k)$ is the intra-cluster distance of cluster X_k.
2. *Davies-Bouldin index* (DBI) is an internal evaluation scheme, where the validation of how well the clustering has been done is made using quantities and features inherent to the dataset. The lower the *DBI index* value, the better is the clustering. It also has a drawback. A good value reported by this method does not imply the best information retrieval. The DBI index for k number of clusters is defined as eq.4:

$$DB(u) = \frac{1}{k}\sum_{i=1}^{k} max_{i\neq j}\{\frac{\Delta(X_i) + \Delta(X_j)}{\delta(X_i, X_j)}\} \tag{4}$$

3. *CPCC* (cophenetic correlation coefficient) index measures the correlation between respective cophenetic matrix P_c and the proximity matrix P of X. Because both matrices are symmetric and have their diagonal elements equal to 0 we consider only the $M = \frac{N(N-1)}{2}$ upper diagonal elements of P_c and P. Let d_{ij} and c_{ij} be the (i, j) element of P and P_c, respectively. Then the *CPCC index* is defined as eq. 5.

$$CPCC = \frac{\frac{1}{M}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} d_{ij}\cdot c_{ij} - \mu_p\cdot\mu_c}{\sqrt{((\frac{1}{M})\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} d_{ij}^2 - \mu_p^2)((\frac{1}{M})\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} c_{ij}^2 - \mu_c^2)}} \tag{5}$$

   $\mu_p = \frac{2}{N(N-1)}\sum_{i=1}^{N}\sum_{j=i+1}^{N} d_{ij}$ and $\mu_c = \frac{2}{N(N-1)}\sum_{i=1}^{N}\sum_{j=i+1}^{N} c_{ij}$. The values of the *CPCC* are between -1 and +1. The closer the *CPCC* index to 1, the better the agreement between the cophenetic and the proximity matrix.
4. *Sillhouette index* measures how well an observation is clustered and it estimates the average distance between clusters. For data point $i \in C$ let $a(i) = \frac{1}{|C_i|1}\sum_{j\in C, j\neq i} d(i, j)$ be the mean distance between i and all other data points in the same cluster, where $d(i, j)$ is the distance between data points i and j in the cluster C_i. We can interpret $a(i)$ as a measure of how well i is assigned to its cluster (the smaller the value, the better the assignment). We then define the mean dissimilarity of point i to some cluster C as the mean of the distance from i to all points in C (where $C \neq C_i$). For each data point $i \in C_i$ we now define $b(i) = min_{k\neq i}\frac{1}{|C_k|}\sum_{j\in C_k} d(i, j)$ to be the smallest (hence the *min* operator in the formula) mean distance of i to all points in any other cluster, of which i is not a member. The cluster with this smallest mean dissimilarity is said to be the "neighboring cluster" of i because it is the next best fit cluster for point i. We now define a *Silhouette (value)* of one data point i (see eq. 6):

$$s(i) = \frac{b(i)a(i)}{max\{a(i), b(i)\}} \texttt{ if } |C_i| > 1 \texttt{ and } s(i) = 0 \texttt{ if } |C_i| = 1. \tag{6}$$

   From the above definition it is clear that -1 $\leq s(i) \leq$ 1. Also, note that score is 0 for clusters with size equal to 1.

The three best global criteria for assessing the quality of grouping are as follows:

1. *Calinski and Harabasz index* (Pseudo F) (eq. 7):

$$G1(u) = \frac{tr(B) \setminus (u-1)}{tr(W) \setminus (n-u)} \tag{7}$$

B is the inter-class covariance matrix, W - the inter-class covariance matrix, tr - the matrix trace, u - is the number of classes and n is the number of objects to be grouped.

2. *Hubert and Levine index* (Hubert index) (eq. 8):

$$G2(u) = \frac{D(u) - r \cdot D_{min}}{r \cdot D_{max} - r \cdot D_{min}} \tag{8}$$

$D(u)$ is the matrix of all inter-class distances, r - number of inter-class distances, D_{min} - smallest inter-class distance, D_{max} - longest inter-class distance.

3. *Gamma Baker and Hubert index* (Gamma index) (eq. 9):

$$G3(u) = \frac{s(+) - s(-)}{s(+) + s(-)} \tag{9}$$

$s(+)$ is the number of compatible distance pairs and $s(-)$ is the number of incompatible distance pairs. If the intra-class distance is smaller than the inter-class distance then we consider the pair as compatible. Such comparisons are $r \cdot c$, where r is the number of inter-class comparisons and c is the number of intra-class comparisons.

6. Rule clustering and quality in R

As we noted in the introduction of this article when we cluster the rules in a given knowledge base and such knowledge base contains outliers in rules this fact definitely influences the cluster quality and then also the efficiency of rule exploration process. To analyze the force of this influence we implement a script in R using two dedicated packages: $stats$ and $clusterSim$. The $stats$ package and the $hclust$ function contained in it enabled hierarchical agglomeration grouping, while the $ClusterSim$ package provided functions to assess the quality of clusters.

6.1. Stats package

The stats package allows for hierarchical cluster analysis on a set of dissimilarities and methods for analyzing it. By using the formula $hclust(d, method = "complete", members = NULL)$ it is possible to use agglomeration method. The possible options are as follows: "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC) or "centroid" (= UPGMC) [3]. Two different algorithms are found in the literature for Ward clustering: "ward.D" without Ward's (1963) clustering criterion [3] implementation and "ward.D2" which implements that criterion [1].

6.2. ClusterSim package

The $cluster.Sim$ package allows for determination of optimal clustering procedure for a data set. Determination of optimal clustering procedure for a dataset by varying all combinations of normalization formulas, distance measures, and clustering methods. By using the formula $cluster.Sim(x, p, minClusterNo, maxClusterNo, icq = "S", outputHtml = "", outputCsv = "", outputCsv2 = "", normalizations = NULL, distances = NULL, methods = NULL)$ in which the most important parameter for us is icq for choosing the internal cluster quality index from the following options: "S" - Silhouette,"G1" - Calinski & Harabasz index, "G2" - Baker & Hubert index ,"G3" - Hubert & Levine index, and others. Other parameters in this formula are explained in [4].

[3] The Ward.D2 criterion values are on a scale of distances whereas the Ward.D criterion values are on a scale of distances squared.

7. Experiments

7.1. Description of the research

The first stage is to load data and organize it. Data subjected to hierarchical clustering must be a numeric matrix, whose rows represent individual observations while columns represent variables. In order to do this the original knowledge bases with rules containing various number of premises are subject to transformation into a matrix in which the rows prompt the rules and the columns all the possible attributes in the premises of the rules with conditional attributes. Such a matrix allows determining similarities between rules and combining similar rules into clusters. For the obtained hierarchical structure, we run a quality analysis of such structure taking into account seven different indicators: *Dunn*, *Davies-Bouldin*, *Sillhouette*, *CPPC* index etc (see Section 5). In our assumptions, we want to know the unusual rules constituting respectively 1, 5 and 10% of all rules in the knowledge base. The three methods: LOF, COF and $KMeans$, analyzed only overlap to some extent in the results, i.e. only some of the rules have identified all three methods as deviations. The subject of a separate study will be the analysis of the similarity of selected methods of detecting deviations, i.e. how similarly they indicate unusual rules. After removing the unusual rules, the quality of the clusters should improve. To check this, we re-calculate previously calculated cluster quality indicators and check in how many cases the quality of clusters has improved, in how many changes in cluster quality were not observed and in how many cases the cluster quality indicator did not react at all to the atypical rules. It turns out that despite the removal of a significant number of rules in the set, the quality of the clusters created does not improve. In this way, it will be possible to analyze the specifics of these indicators and what really affects their values. Ultimately, it will also turn out that a specific set of data has a very large impact on the results obtained in this experiment. There are data that did not allow to improve the quality of clusters despite the removal from the collection of a significant number of clusters. Assessment of the nature of these collections will be the subject of separate studies. There are six different knowledge bases on which we based the research: *weather*, *diseases*, *libra*, *diabetes*, *nursery* and *krukenberg*. 5 of them (weather, diseases, libra, diabetes and nursery) are from a known repository *UC Irvine Machine Learning Repository* [7]. The sixth dataset, *krukenberg* is a real-life knowledge base created for medical domain. The process of rules acquisition from the original dataset is as follows. The authors focus on knowledge representation in the form of rules generated automatically from data with the use of *rough set theory* and the $LEM2$ algorithm, proposed by [8], implemented in $RSES$ system. It generates short rules, easily interpretated. Of course, there are many rules induction algorithms from data: generating decision rules from decision trees and algorithms for generating association rules. When the size of input data (the ones that rules are to be generated from) increases, the number of generated rules does too. *diabetes* data set [7] contains 768 of objects described with 8 continuous attributes. The objects are divided into two decision classes where 1 means „tested positive for diabetes” and covers 268 objects and 0 means the opposite and covers 500 instances. Processing the data with $LEM2$ and $RSES$ which contains an implementation of the $LEM2$ algorithm, 490 rules have been created. The short characteristics of each knowledge base are presented in the Table 1. They differ with number of rules, number of attributes but also the type of the attribute, the length of the rules.

Table 1. An example of a table.

KB	weather	diseases	libra	krukenberg	diabetes	nursery
$\#rules$	5	99	165	200	490	867
$\#attributes$	5	14	91	23	9	9

The course of performed experiments can be described as follows.

1. The following tests were performed for each of the 6 knowledge bases (KBs).

   (a) For each of the 7 methods of clusters combining (*single*, *complete*, *average*, *centroid*, *mcquitty*, *ward* and *ward*2) a $CPCC$ coefficient was determined. This factor answers the question to what extent dendrogram reflects the distance between pairs of data from a given set, i.e. the degree of matching of the dendrogram to the distance matrix. In this way, we obtain information about the method of combining clusters that allows you to obtain an optimal cluster distribution.

(b) For the optimal CPCC found, the optimal number of clusters was searched, starting from $k = 2, 3, \ldots$ increasing the step by 1 to 10.
(c) For the optimal number of clusters found, all cluster quality indices were calculated (except *CPCC* we have 6 other indices: Dunn, Davies-Bouldin, etc.).

2. Only results obtained for optimal divisions and cluster quality indicators were taken for further analysis.
3. For each individual experiment, the optimal solution was searched for using the *LOF*, *COF* and *Kmeans* algorithms for 1%, 5% and 10% deviations, respectively.
4. The indicated outliers were removed from the knowledge base and cluster quality indicators were determined again.

7.2. Results of the experiments

The subsection contains the results of performed experiments. Tables and charts apply only to selected aspects examined by the authors. As mentioned above, for each of the algorithms for outliers detecting in the rules: *LOF*, *COF* and *Kmeans*, 1%, 5% or 10% of outliers were determined. These outliers (unusual rules) were removed from the knowledge base and it was checked in which cases the quality of clusters improved, deteriorated and no changes in cluster quality were recorded at all. Table 2 presents the results achieved for every of seven used quality indexes (*Silhouette*, *Dunn*, *Davies-Bouldin*, etc.) and each of the three used outlier mining algorithms: *LOF*, *COF* and *Kmeans*. It shows the frequency of improving / deterioration or no quality changes in rules clustering after outliers removing.

Table 2. The influence of quality indices and outlier detection method on the frequency of clusters' quality improvement or deterioration.

Quality indices	Outlier detection method	improvement of quality	deterioration of quality	no quality changes
Silhouette	LOF	88, 90%	11, 10%	0%
	COF	83, 30%	16, 70%	0%
	KmeansS	72, 20%	22, 20%	5, 60%
Dunn index	LOF	16, 70%	0%	83, 30%
	COF	83, 30%	0%	16, 70%
	Kmeans	83, 30%	0%	16, 70%
Davies - Bouldin index	LOF	100%	0%	0%
	COF	100%	0%	0%
	Kmeans	100%	0%	0%
Pseudo F	LOF	100%	0%	0%
	COF	100%	0%	0%
	Kmeans	100%	0%	0%
Hubert index	LOF	55, 60%	44, 40%	0%
	COF	55, 60%	44, 40%	0%
	Kmeans	38, 90%	44, 40%	16, 70%
Gamma index	LOF	83, 30%	0%	16, 70%
	COF	83, 30%	0%	16, 70%
	Kmeans	72, 20%	11, 10%	16, 70%
CPCC	LOF	100%	0%	0%
	COF	100%	0%	0%
	Kmeans	100%	0%	0%

It shows that there are some indexes (like *Davies - Bouldin*, *Pseudo F* and *CPCC*) for which the quality of rule clusters always improve after removing at least one rule. It means that these indexes are very sensitive to the outliers in a given dataset. In case of using the *Silhouette index* or *Hubert and Levine index* sometimes we observed the deterioration of quality after removing the unusual rules. This case requires further research.

As it was mentioned before, we also decide to review the experiments in which we removed 1%, 5% and 10% of unusual rules from a given knowledge base. We wanted to check whether the frequency of improving the quality of rule clusters depends on the number of outliers and the chosen quality indexes. The results are included in Table 3.

Table 3. The frequency of improving the quality of rule clusters vs. number of outliers.

%outliers	Silhouette	Dunn	Davies - Bouldin	Pseudo F	Hubert index	Gamma index	CPCC
1%	83, 3%	61, 1%	100, 0%	100, 0%	44, 4%	72, 2%	100, 0%
5%	83, 3%	61, 1%	100, 0%	100, 0%	50, 0%	83, 3%	100, 0%
10%	77, 8%	61, 1%	100, 0%	100, 0%	55, 6%	83, 3%	100, 0%

We may easily see that no matter how many outliers we remove always the indexes manage to produce improved quality were *Davies-Bouldin*, *Pseudo F* and *CPCC*. The more outliers in rules we detected and removed the higher was the frequency of quality improving.

We were also interested in how often the analyzed outlier detection algorithms (*LOF*, *COF* and *Kmeans*) lead to improvement of cluster quality in general. The results are included in Table 4.

Table 4. The frequency of clusters quality improvement vs. outlier detection method.

	improvement of quality	deterioration of quality	no quality changes
LOF	78%	8%	17%
COF	87%	9%	5%
Kmeans	80, 94%	11, 10%	7, 96%

Surprisingly, the algorithm which the most often results in no quality changes is *LOF* algorithm, one of the most popular outlier detection algorithms. Definitely, the optimal choice in this case would be using *COF* algorithm, which in 87% of all cases results in improving rule clusters quality. In our research we wanted to review the seven quality indexes in case of rules exploration as outliers. We wanted to check which index most frequently improve the clusters quality. It seems (what can be observed in Table 5) that the indexes which defenately are sensitive on outliers apearing in a given knowledge base are *Davies-Boulding*, *Pseudo F* and *CPCC* indexes. The index that least often reacted to the occurrence of outliers in rules is *Hubert and Levine* index.

Table 5. The frequency of clusters quality improvement vs. quality indexes.

Silhouette	Dunn	Davies-Bouldin	Pseudo F	Hubert index	Gamma index	CPCC
81, 5%	61, 1%	100%	100%	50%	79, 6%	100%

The last aspect of our research concerned the nature of the data. We wanted to check if the data type influences the achieved results. Figure 1 shows that there are knowledge bases with rules in which, in each case, after the removal of unusual rules, there has been an improvement in the quality assessment of clusters of rules (in case of *krukenberg*). On the other hand, the knowledge base, which in 70% of cases did not allow to improve the quality of clusters of rules despite the removal of unusual rules was *weather*).

8. Summary

The subject of outlier mining is very important nowadays. Outliers in rules mean unusual rules which are rare in comparison to others and should be explored further by the domain expert. By grouping together rules which are not similar enough, we reduce the quality of clusters and their representatives) and thus contribute to a decrease in the efficiency of inference. For this reason, we became interested in the impact of unusual rules on the quality of clusters. In order to review this we analyzed many different outlier mining algorithms and finally we chose 3 methods: *LOF*, *COF*, *KMeans*. They were previously used for typical simple data. Now we used them to examine the effectiveness of detection of unusual (outlier) rules as well as the sensitivity of selected measures of cluster quality to the presence of outliers in the data. In the research the authors use the outlier detection methods to find a given (1%, 5%, 10%)

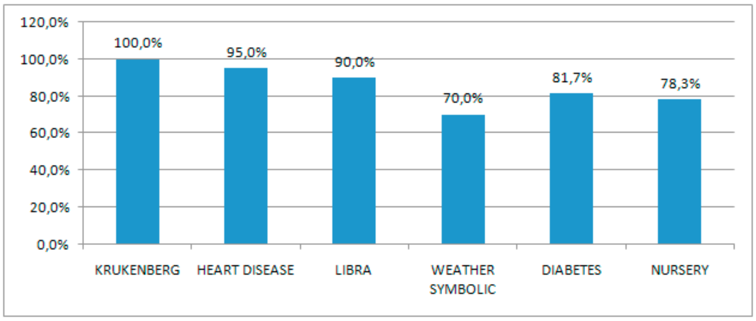

Fig. 1. Knowledge base vs. quality of rule clusters improvement

number of outliers in rules. Then, they analyze which of seven various quality indices, that they used for all rules and after removing selected outliers, improve the quality of rule clusters. In the experimental stage the authors used six different knowledge bases.

The results show that the most optimal results were achieved for the *COF* outlier detection algorithm as the one for which, for all analyzed quality indices, the most often the cluster quality was improved. There are some indexes (like *Davies - Bouldin*, *Pseudo F* and *CPCC*) for which the quality of rule clusters always improve after removing at least one rule. It means that these indexes are very sensitive to the outliers in a given dataset. The index that least often reacted to the occurrence of outliers in rules is *Hubert and Levine index*. What is more, the nature of the data also influences the rule clusters' quality. The knowledge base for which, always, after removing unusual rules we observed an improvement in the rules quality is *krukenberg*. The one which, in 70% of cases, did not allow to improve the quality of clusters of rules despite the removal of unusual rules was the *weather* knowledge base.

References

[1] Legendre, Pierre, and Fionn Murtagh (2014) „Wards Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Wards Criterion?" *Journal of Classication* **31**: 274–295.

[2] Walesiak, Marek (1993) „Statystyczna analiza wielowymiarowa w badaniach marketingowych" [in polish], *Prace Naukowe AE we Wrocławiu nr 654*, Seria: Monografie i Opracowania nr 101, AE, Wrocław.

[3] https://stat.ethz.ch/R-manual/R-devel/library/stats/html/hclust.html R Stats Package,in R version 2.15. Based on code by F. Murtagh (Murtagh 1985), included in R by Ross Ihaka and Fritz Leisch.

[4] Walesiak, Marek, Dudek, Andrzej. (2017) http://keii.ue.wroc.pl/clusterSim/.

[5] Gower, John C. (1971) „A general coefficient of similarity and some of its properties" *Biometrics*.

[6] Wierzchoń, Sławomir, Kłopotek, M.A. (2015) „Algorytmy analizy skupień". [in polish], *WNT*, Warszawa.

[7] https://archive.ics.uci.edu/ml/index.php

[8] Grzymała-Busse Jerzy (1997) „A new version of the rule induction system LERS", *Fundam. Inform.* **31 (1)** 27-39. https://doi.org/10.3233/FI-1997-3113.

[9] Nowak-Brzezińska, Agnieszka, Wakulicz, Alicja (2019) „Exploration of rule-based knowledge bases: A knowledge engineers support", *Information Sciences*, **485**: 301-318, Elsevier

[10] Nowak-Brzezińska, Agnieszka (2018) „Enhancing the efficiency of a decision support system through the clustering of complex rule-based knowledge bases and modification of the inferencje algorithm", *Complexity*, **2018**, Article ID 2065491.

[11] Grubbs, Frank (1969) „Procedures for Detecting Outlying Observations in Samples", *Technometrics*, **11**, (1):121, doi:10.1080/00401706.1969.10490657.

[12] Goldstein Markus, Uchida Seiichi (2016) „A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data", *PLoS One*, **2016**, 11(4):e0152173, doi:10.1371/journal.pone.0152173.

[13] Breunig, Markus M., Kriegel,Hans-Peter, Ng, Raymond T., Sander, Jrg (2000) „LOF: Identifying Density-Based Local Outliers". *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, 93104, Dallas, Texas, USA.

[14] Tang, Jian, Chen, Zhixiang, Fu, Ada, Cheung, David (2002) „Enhancing eectiveness of outlier detections for low density patterns". *Advances in Knowledge Discovery and Data Mining*, **LNCS 2336**, 535548. Springer Berlin / Heidelberg.

[15] Hartigan, John, Wong, M. A. (1979) „A K-Means Clustering Algorithm", *Applied Statistics*, **28**:1, 100-108.

[16] Mennatallah, Amer, (2011) „Comparison of Unsupervised Anomaly Detection Techniques", *Bachelor Thesis*, Media Engineering and Technology German University in Cairo.