

Outlier detection for categorial data using clustering algorithms

Agnieszka Nowak - Brzezińska¹[0000–0001–7238–1170] and
Weronika Łazarz¹[0000–0002–1509–5909]

Institute of Computer Science, Faculty of Science and Technology,
University of Silesia, Bankowa 12, Katowice 40-007, Poland;
agnieszka.nowak-brzezinska@us.edu.pl

Abstract. Detecting outliers is a widely studied problem in many disciplines, including statistics, data mining and machine learning. All anomaly detection activities are aimed at identifying cases of unusual behavior when compared to the remaining set. There are many methods to deal with this issue, which are applicable depending on the size of the dataset, the way it is stored and the type of attributes and their values. Most of them focus on traditional datasets with a large number of quantitative attributes. While there are many solutions available for quantitative data, it remains problematic to find efficient methods for qualitative data. The main idea behind this article was to compare categorical data clustering algorithms: *K-modes* and *ROCK*. In the course of the research, the authors analyzed the clusters detected by the indicated algorithms, using several datasets different in terms of the number of objects and variables, and conducted experiments on the parameters of the algorithms. The presented study has made it possible to check whether the algorithms detect the same outliers in the data and how much they depend on individual parameters such as the number of variables, tuples and categories of a qualitative variable.

Keywords: qualitative data · outlier detection · data clustering · *K-modes* · *ROCK*.

1 Introduction

The article deals with the clustering of qualitative data to detect outliers in these data. Thus, in the paper, we encounter two research problems: clustering qualitative data and detecting outliers in such data. We look at outliers as atypical (rare) data. If we use clustering algorithms for this purpose, outliers are data that are much more difficult to include in any group than the typical (normal) data. Clustering qualitative data is a more extensive research problem than clustering quantitative data. We count the distance between the numeric values on each attribute that describes the objects. Quantitative data can be normalized which allows us to interpret the differences between the compared objects properly. Assessing the similarity between two objects described by qualitative attributes is a challenging task. Let us take *eye color* as an example of a qualitative attribute. Now, let us take into account three persons: *A* with *blue* eyes, *B* with *brown* and *C* with *gray* eyes. There are various methods to measure their similarity. We may say that *blue* is more similar to *gray* than *brown*. In fact, we know that *gray* is much

more similar to *blue* than *brown*. But we may also want to compare them as a plain text and then *blue* and *brown* share the same initial letter which makes them more similar than pairs *blue-gray* or *brown-gray*. It all depends on the method we use to compare the objects. It is also worth remembering that the comparing of the objects in the set will significantly impact the structure of the groups that we create.

By default, clustering algorithms, known in the literature for years, are based on the concept of data distances in a metric space, e.g., in Euclidean space. The smaller the distance between the objects, the greater the probability that they will form one group. If the distance between a given object from all created groups is too great, then we should consider the object as an *outlier* in the data. This idea seems logical. In the context of qualitative data: when a given object shows no similarity to the created groups, then it can be considered an outlier in the data.

In the study, we have made use of real datasets from various fields. This type of data very often contains some unusual pieces of data. They are not the result of a measurement error, but they actually differ from most of the data in the set. It is not always the case that one or more objects stand out significantly from the rest, and we can easily see it. Sometimes, it is also the case that certain subsets of objects differ to the same extent from the majority of data. The problem becomes even more complicated when we take into account the fact that these objects in the sets may be more or less differentiated by the specificity of the domain they come from, but also by the method of describing these data (the number of attributes, the number of possible values of these attributes, the number of objects). When objects are described on a categorical scale, the effectiveness of their correct clustering and outlier detection is necessary for a deeper study. In this paper, we analyze clustering algorithms from two types of clustering: hierarchical (*ROCK*) and non-hierarchical (*K – modes*). In case of quantitative data, the clustering process works as follows. Hierarchical algorithms in each iteration look for a pair or groups of objects with a smallest distance and combine them into a group. The process is repeated until an expected number of clusters is reached or until all groups have merged into one group. On the other hand, non-hierarchical algorithms (like the most popular clustering algorithm *K – means*), search for the best partition for a predetermined number of groups so that the distances inside the clusters are small and the clusters are as large as possible. In qualitative data, we should modify the algorithms to be suitable for operating on data for which we cannot explicitly measure distances. In case of non-hierarchical algorithms, we cannot use the *K – means* algorithm because it forms its representative by determining the value of the so-called center of gravity of the group. For quantitative data, it is simply an arithmetic mean of the attribute values describing the features that make up the group. For qualitative data, we cannot derive a mean value. However, we can find a most common value. And this is the concept behind the *K – modes* algorithm we chose for our research. In case of hierarchical algorithms, where two objects with the shortest distance are combined into a group iteratively, for datasets with qualitative data we cannot rely on the notion of distance. Instead, we use measures to determine the similarity of objects and, at each step of the algorithm, we connect the objects or groups of objects with the greatest similarity. This is the main idea of the *ROCK* algorithm - a hierarchical clustering algorithm for qualitative data. We group the data to explore it better. Exploration has to do with the

fact that apart from its obvious task, which is discovering patterns or rules in data, we can also discover unusual data, outliers in the data.

Therefore, in this study, we decided to investigate the effectiveness of the two selected clustering algorithms: *K – modes* and *ROCK*, in outlier detecting. We want to compare how consistent the algorithms are in this respect. If they are consistent, then they should designate the same objects for potential outliers. In the research, we will change the clustering parameters to find the optimal results. We will repeat the experiments for 5%, 10%, and 15% outliers in the dataset. We expect that the more outliers we identify, the greater the coverage of the analyzed methods may be. We present the results in the section on experiments and research results.

2 State of art

The methods of outlier detecting in datasets can be divided into formal and informal. Most formal tests require test statistics to test hypotheses and usually rely on some well-behaved distribution to check whether the extreme target value is out of range. However, real-world data distributions may be unknown or may not follow specific distributions. That is why it is worth considering other solutions, for example, clustering algorithms. In addition to the distribution-based methods, cluster-based approaches are also welcome. These approaches can effectively identify outliers as points that do not belong to the created clusters or the clusters distinguished by a small number of elements [9, 6]. So far, numerous works have been published focusing on detecting outliers and good data clusters in a quantitative dataset. The most well-known algorithm is the *LOF* (Local Outlier Factor) algorithm proposed by Breunig in [2], in which local outliers are detected. Based on the ratio of the local density of a given object and the local density of its nearest neighbors, the *LOF* factor is calculated. Then, the objects with the highest *LOF* values are considered as outliers. Another method that isolates outliers and normal objects is the *IsolationForest* method based on the construction of a forest of binary isolation trees. Then outliers are observations with shortest average path lengths from the root to the leaf [8]. The indicated algorithms are widely used in IT systems, both to clean datasets from noise so that they do not interfere with the system operation, and to detect unusual observations in the data for a further analysis. The presence of outliers in qualitative data can significantly disrupt the effectiveness of machine learning algorithms that try to find patterns in the data, such as rules, decision rules or association rules. Dividing the objects into groups in which the objects are as similar to each other as possible and thus detecting objects that do not match the groups is a very efficient solution to explore the outliers. We decided to choose two clustering algorithms, *K – modes* and *ROCK* - as they are the representatives of both hierarchical and non-hierarchical clustering algorithms. We found them very simple to interpret and implement on real data. So far, no papers describing the application of the indicated algorithms on a large scale or comparing the results with the distinction as to the type of data processed and the time of execution have been published. This has become the direct motivation of the authors of this paper to analyze those two selected clustering algorithms *K – modes* and *ROCK* in the context of their efficiency in detecting outliers in the qualitative data.

3 Data clustering

The problem of clustering is one of the most researched issues in social sciences, psychology, medicine, machine learning, and data science. In addition to the standard benefits of data clustering, it has found a wide application in dataset processing with categorical domains, both in the course of preparation for mining and in the modeling process itself. Here, data clustering was used to find outliers in qualitative datasets. The two algorithms described in this section differ in terms of data clustering and outliers detection. The *K – modes* algorithm, most frequently used in research and real IT systems, creates groups of clusters from objects closest to selected centroids and defines outliers as objects farthest from the cluster center. The *ROCK* algorithm calculates the similarity measures between objects and groups of objects, creating data clusters containing objects that should not belong to any other cluster.

When dealing with quantitative data, we can easily use descriptive statistics, using quantities such as mean, median, standard deviation, and variance. When we handle qualitative data, it is not possible. We only know the most common value - a dominant. In such a case, clustering algorithms will cluster objects with the same value of a given attribute into groups. Of course, large clusters will be created by objects with a value equal to the dominant for a given attribute. For the clusters to be of good quality, we must effectively detect unusual data not to disturb the coherence and separation of the created data structures. We do not make assumptions that our sets contain outliers. We want our model to deal with any given dataset. If there are no outliers in the set, the cluster quality indicators will be very close to the values expected for the sets without outliers.

3.1 K-modes clustering

The *K – modes* clustering algorithm was proposed as an alternative to the popular *K – means* algorithm, the most used centroid-based non-hierarchical algorithm [5]. The modifications made to the *K – means* algorithm include using a simple measure of matching dissimilarity for qualitative features, replacing the group averages with vectors composed of the most common values at individual coordinates of the objects (modes), and using a frequency-based method to modes update. Let $X = \{x_1, \dots, x_n\}$ be a set of n -objects x , such that $x = (x_1, \dots, x_m)$. The dissimilarity measure of x_1, x_2 objects is defined as $d(x_1, x_2) = \sum_{i=1}^m \sigma(x_{1i}, x_{2i})$, where $\sigma(x_{1i}, x_{2i}) = 0$ if $x_{1i} = x_{2i}$ and 1 otherwise. Having $A = \{A_1, \dots, A_m\}$ - set of the attributes of the objects in X it is possible to define $S \subseteq X$ - a cluster of data. The mode of $S = \{x_1, \dots, x_p\}, 1 \leq p \leq n$ is the vector $q = (q_1, \dots, q_m)$ which minimizes the function $D(S, q) = \sum_{i=1}^n d(x_i, q)$ called the cost function. A cluster center is called a mode and is defined by considering those values of the attributes that appear most frequently in the data points which belong to that cluster. The *K – modes* (Algorithm 1) algorithm begins with a random selection of k objects (centroids) which are the central objects of k clusters. Then, the dissimilarity measure is calculated and the closest centroid is determined for each object. When all objects are assigned to individual clusters, the centroids are updated by creating new modes from objects present in the cluster. The calculations are repeated until the differences in the generated clusters in the following steps cease to exist.

Algorithm 1 $K - modes$ algorithm

input: X -dataset, k -expected number of clusters*output*: a set of k clusters

1. Randomly select k items (modes) from the dataset.
 2. For each pair (mode, object), calculate the dissimilarity measure.
 3. For each object that is not a mode, find the mode closest to the object.
 4. Join objects with the corresponding modes to create clusters.
 5. For all clusters, recalculate the modal vectors containing in successive coordinates the most common values on attributes of cluster objects.
 6. Perform steps 3-5 until the generated clusters do not repeat themselves.
-

The $K - modes$ algorithm is the easiest to implement and the most popular among the categorical data clustering algorithms because it is linearly scalable concerning the size of the dataset. The disadvantage of the algorithm is that it selects random initial modes, leading to unique structures around objects that are undesirable in the set. A method to prevent such situations is to draw the initial set of modes multiple times and assign each object to the cluster with the greatest number of times. The output clusters generated by the $K - modes$ algorithm have a similar cardinality, which does not have to reflect the actual data clusters on the sets having atypical distributions of variables. As with most categorical clusters, clusters containing a tiny number of elements or a single element can be considered outliers. The specifics of $K - modes$ clustering show that we will create single-element clusters only if the initially drawn object is an outlier. If we want to obtain a reliable mapping in small individual clusters, we can run the algorithm multiple times, each time randomizing a different set of initial $K - modes$ and finish the work when the variability is low in the final set of clusters. Finding the similarity between a data object and a cluster requires n operations, which for all k clusters is nk . Assigning objects to the appropriate k clusters and updating mods also require nk operations. Assuming the algorithm is run I times for different starting objects, the algorithm will have a linear complexity of $O(nkI)$.

3.2 ROCK Clustering

The *ROCK* algorithm (RObust Clustering using linKs) [4], is a hierarchical clustering algorithm for categorical data. The algorithm introduces notions of neighbors and links. A point's neighbours are those points that are considerably similar to it. A similarity function between points defines the closeness between pairs of points. A user defines the threshold for which the pairs of points with a similarity function value greater than or equal to this value are considered to be neighbors. The number of links between pairs of points is defined to be the number of common neighbors for the points. The larger the number of links between a pair of points, the greater the likelihood is that they belong in the same cluster. Starting with each point in its own cluster, the algorithm repeatedly merges the two closest clusters till a desired number of clusters remain or when a situation arises in which no two clusters can be merged.

Algorithm 2 *ROCK* algorithm

input: sample set of objects. Number of k clusters to be found. The similarity threshold: $\theta \geq 0.4$

output: A group of objects - a cluster

Do for All Data {

1. Initially, place each object into a separate cluster.
2. Construction of a Similarity Matrix with similarity for each pair of objects (A,B) using measure $Similarity(A, B) = \frac{|A \cap B|}{|A \cup B|}$
3. Computation of an Adjacency Matrix (A) using a similarity threshold $\theta \geq 0.4$ if $similarity(A, B) \geq \theta$ then 1; else 0
4. Compute a Link Matrix by multiplying an Adjacency Matrix by itself to find the number of links.
5. Calculation of a Goodness Measure for each pair of objects by using the g function
6. Merge the two objects with the highest similarity (goodness measure).
7. When no more entry exists in the goodness measure table then stop the algorithm which by now should have returned k number of clusters and outliers (if any), otherwise go to step 4.

}

The following features of this algorithm are necessary to define:

- Links - the number of common neighbors between two objects.
- Neighbors - if a similarity between two points exceeds certain similarity threshold, they are neighbors: if $similarity(A, B) \geq \theta$ then two points A, B are neighbors, for θ being a user-specified threshold.
- Criterion Function - the objective is to achieve a good cluster quality by maximizing the sum of links of intra cluster point pairs and minimizing the sum of links of inter cluster point pairs.
- Goodness Measure to maximize the criterion function and identify the best pair of clusters to be merged at each step of the *ROCK* clustering algorithm.

ROCK is a unique algorithm because it assumes that an attribute value, in addition to its frequency, must be examined based on the number of other attribute values with which it occurs. Due to its high computational complexity, *ROCK* is good at detecting outliers in small datasets, and its computational time increases as the records in the set increase. This is because each record must be treated as a unique data cluster. If the user does not have a comprehensive knowledge about the dataset, the appropriate selection of the θ value and the minimum number of clusters generated on the output is a challenging task. The *ROCK* algorithm is very resistant to outliers and can successfully identify outliers that are relatively isolated from the rest of the points. The ones with very few or no neighbors in one- or several-member clusters will be considered outliers. The overall computational complexity will depend on the number of neighbors of each facility. In most cases, the order of complexity will be $O(n^2 \log n)$. If a maximum and an average number of neighbors are close to n , then the algorithm's complexity increases to $O(n^3)$.

4 Conducted research

The algorithms described in Section 3 were implemented in the Python language (version 3.8.8). We used the JupyterHub (version 6.3.0) environment available at <https://jupyter.org/hub> for the implementation and visualization of the data. JupyterHub runs in the cloud or on hardware locally and supports a preconfigured data science environment for each user. We used Anaconda package containing most of the libraries, enabling machine learning models and visualization of results. The existing models of the Scikit-Learn library were used to implement the $K - modes$ algorithm. The *ROCK* algorithm due to a lack of previous implementation was implemented by the authors. We used the Matplotlib library and the Pandas Dataframe structure for data visualization. Most of the computation is based on the Pandas data structures that hold the results.

The computer program described by the authors has been divided into sections containing:

- Importing Python libraries SciPy (1.6.2), Scikit-learn (0.24.1), NumPy, Pandas (1.2.4), Matplotlib (3.3.4) and libraries to perform operations related to time.
- Implementing algorithms: *ROCK* with the parameters: k denoting the expected number of clusters and θ being a parameter of a function that returns an estimated number of neighbors and $K - modes$ with k parameter denoting the expected number of clusters and $threshold$ parameter denoting the percentage of expected outliers.
- Data preprocessing: dealing with missing values (function that completes missing fields with the most common value in a column and removes columns that contain more than 60 empty values), coding the variables (encoding text values into numerical values), decoding encoded text variables.
- Uploading all datasets (reading, calculating the descriptive statistics, encoding text variables for the selected dataset to visualize the result).
- Execution of *ROCK* and $K - modes$ algorithms on datasets. Presentation of the algorithms' computation time in relation to the type of the algorithm.
- Presentation of the algorithms' computation time in relation to the number of variables, the number of records, and data diversity.
- Listing the numbers of individual clusters obtained by the *ROCK*, $K - modes$ algorithms.
- Showing the selected dataset with assigned cluster numbers for the *ROCK* and $K - modes$ algorithms and flags that indicate whether a record has been classified as an outlier. If the flag is -1 , the object is an outlier. If it is 1 , the object is considered normal.
- Presentation of the matrix of similarities and differences in classifying values as outliers for the *ROCK* and $K - modes$ algorithms when compared in pairs.
- Identification of common outliers generated by the *ROCK* and $K - modes$ algorithms.

The source of the software was placed in the GitHub repository: <https://github.com/wlazarz/outliers2>. It contains the implementation of the $K - modes$ and *ROCK* algorithms and six datasets on which the experiments were conducted. The sequence of steps performed to compare the clustering and outliers detection algorithms is presented in Figure 1. The equipment specification on which we conduct our research

is as follows: MacBook Pro Retina (15-inch, Mid 2015), macOS Catalina (10.15.7), 2,2 GHz processor four-core Intel Core I7, RAM 16 GB 1600 MHz DDR3, GPU Intel Iris Pro 1536 MB. GPU acceleration and XAMPP were not used.

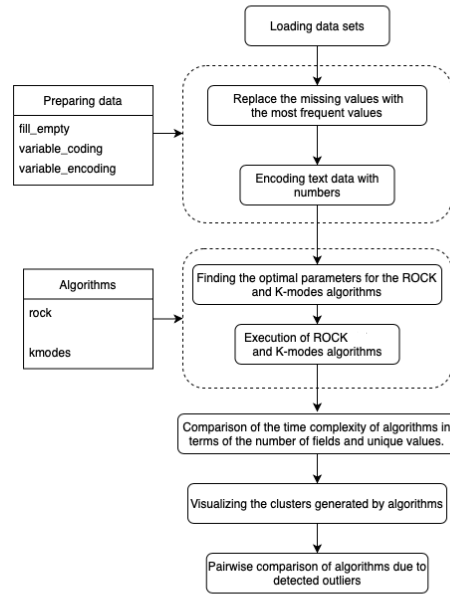


Fig. 1. Scheme of the program comparing algorithms clustering data and detecting outliers.

4.1 Data description

We used six qualitative datasets to compare the algorithms that detect outliers in the data, each with a different structure of the variables matched to the clustering-based algorithms which support the detection of outliers in the qualitative datasets. The sets have different sizes and consist of a different number of categorical variables. The characteristics of the selected datasets are presented in Table 1. All analyzed datasets are real datasets, four of which relate to the domain of medicine (*Primary Tumor* [10], *Lymphography* [11], *SPECT Heart* [12], *Covid – 19* [13]). In addition to the medical databases, two others were also analyzed: *BM_attack* [16] and *wiki* [15]. The set *wiki* contains the highest number of objects (913) and attribute values (285 unique values).

The first step in the project was to load datasets and prepare them properly before clustering commences. In all datasets, we filled empty fields with the most common value on a given variable. Categorical variables were encoded into numeric variables on *Primary Tumor Dataset* and *Lymphography Dataset*. Despite reducing the dataset to a numerical form, algorithms working on qualitative sets treat numbers as categories of

variables. The process of numerical encoding of the test values was intended to reduce a long execution time of the algorithms resulting from the need to compare each sign of the test value.

4.2 Methodology

We conducted the experiments empirically. Initially, we tried to automate the experiments by launching the execution of the algorithms: *K – modes* and *ROCK*, and changing the parameter values of these algorithms iteratively. However, several lengthy multi-hour processes were interrupted by an excessive memory consumption. As a result, the experiments were finally carried out empirically for the gradually and consciously changed parameter values (e.g., number of clusters). The *elbow method* was used while looking for parameters for the *K – modes* algorithm [14]. If the number of clusters selected with this method generated substantial outliers (many objects were on the border of 5%, 10%, 15% of outliers), the number of clusters was increased or decreased, still oscillating around the threshold point. The authors checked a cluster relevance using the *Silhouette method*, but the structure of created clusters was not always satisfactory [17]. In case of the *ROCK* algorithm we took into account the number of clusters (already established during the execution of the *K – modes* algorithm) and an initial epsilon value (a maximum distance at which elements can be in one cluster) = 0.6. Most of the sets we dealt with had a reasonable number of outliers within the epsilon value of 0.6. If too many outliers were obtained, the epsilon value was increased. If increasing this value results in even more outliers, the number of clusters was decreased. Conversely, for too few outliers obtained, the epsilon was reduced, or the number of clusters was increased.

5 Experiments

This section covers the results of the comparison of the two algorithms described in the previous section: *ROCK* and *K – modes*. We compared the algorithms in terms of their time complexity. At the very beginning, it is worth emphasizing that in this paper, we present the results obtained as a result of optimization of clustering parameters. Thus, by diligently changing the clustering parameters of both algorithms, we checked which combination of the values of these parameters gives optimal results. These optimal results (as one of many obtained) are presented below.

5.1 Time complexities of clustering algorithms

Based on the sets described in Section 4.1, we performed an analysis of time complexity of the algorithms described in this work. The execution time of the algorithms is given in seconds. The study was conducted in the *JupyterHub* environment installed locally on *MacBookPro* hardware with *IntelCorei7* quad-core processor and 16 GB RAM. The datasets are characterized by a different number of objects and variables and represent different types of data. The results are included in Table 1.

Table 1. Time complexity for *ROCK*, *K – modes* and *K – means* clustering algorithms

Dataset	Rows	Columns	Values	Time Complexity [s]		
				ROCK	K-modes	K-means
BM_attack	322	6	20	5,81	1,4	0,11
SPECT	267	23	46	3,67	2,91	0,47
primary-tumor	339	18	58	6,91	3,18	0,77
lymphography	148	19	62	0,72	1,52	0,22
covid	204	16	91	1,64	1,84	0,29
wiki	913	53	285	141,96	26,57	3,06

The *K – modes* algorithm has an average linear or near-square complexity when diagnosed with many clusters. Regardless of the number of records, variables, and values, the execution time for the *K – modes* algorithm is the lowest for each dataset. We can observe that the complexity of the *ROCK* algorithm increases rapidly with the increase in the number of data.

5.2 Outlier detection for clustering results

Algorithms working on qualitative datasets require the indication of individual parameters for the dataset: the number of generated clusters in case of the *K – modes* algorithm and a minimum number of generated clusters and in case of the *ROCK* algorithm the estimated number of neighbors between objects in the clusters. Implementing the *ROCK* algorithm became a tough challenge due to a very high computational complexity and unusual parameters. We selected the *ROCK* algorithm parameters on a trial and error basis. While the *ROCK* algorithm analyzes the similarities not only between objects but also between clusters that should be merged into a single cluster, the *K – modes* algorithm arranges objects from a dataset between clusters so that each cluster contains a similar amount of data and focuses only on the similarities between individual objects in the data. As mentioned earlier, the definition of an outlier generated by the *ROCK* algorithm, taken from [4] indicates one-element classes. The records marked as anomalies by the *K – modes* algorithm are the records from the farthest neighborhood of the centroid in which cluster the object is located. All datasets used in this research were taken from the *UCI Machine Learning Repository* database and represent real data collected during research on real data objects with different distributions, possibly containing a small number of deviations, which results in significantly different sizes of clusters generated by the *ROCK* algorithm. The results of the outlier detection analysis for the lymphography set are presented in Figure 2.

Data clustering algorithms do not have a natural definition of outliers and do not return points considered as variances in the data. The problem of marking objects that differ the most from the others due to the calculations characteristic of the algorithm was solved by generating an additional column for the dataset containing the values -1 or 1 , where the value -1 means that the object was considered an outlier and 1 means that the object is normal. In most cases, the analyzed algorithms returned completely different results. Large differences in outliers selection are the results of the different nature of those algorithms. The *ROCK* algorithm is the most diligent in

Class	lymphatics	block of affere	bl. of lymph. c	bl. of lymph. s	by pass	extravasates	regeneration of	early uptake in	lym.nodes dimin	lym.nodes enlar	changes in lym	defect in node	changes in node	changes in stru	special forms	dislocation of	exclusion of no	no. of nodes in	ROCK clusters	K-modes clusters	ROCK labels	K-modes labels	
14	1	3	2	2	2	2	2	2	3	1	1	2	2	8	1	2	2	4	0	5	-1	-1	
41	1	2	2	2	1	2	2	2	1	4	2	2	2	4	3	2	2	6	0	2	-1	-1	
44	1	3	2	2	2	2	2	1	2	2	2	4	2	4	3	2	2	7	0	5	-1	-1	
72	1	4	2	1	1	2	2	1	1	1	3	3	3	3	3	2	2	3	0	5	-1	-1	
90	1	4	2	2	1	2	2	1	2	1	1	2	2	1	3	1	2	2	0	5	-1	-1	
92	1	3	2	2	1	2	2	1	2	1	2	3	3	4	2	2	2	1	1	0	0	-1	-1
122	1	2	2	2	2	2	2	1	1	1	2	2	4	3	8	2	2	2	3	0	5	-1	-1
136	1	1	1	1	1	1	2	1	2	1	2	2	1	1	2	1	1	1	2	0	1	-1	-1

Fig. 2. The results of the outlier detection analysis for the lymphography set

detecting outliers. It focuses on inter-object and inter-cluster connections, tying them together until well-defined clusters are obtained with the number of common neighbors below a certain threshold. Thus, single-member clusters contain far-away objects from every other cluster and every data object. In case of the $K - modes$ algorithm, due to randomness during the selection of an initial set of cluster centroids, outliers are considered as the objects whose distance from the centroids in the clusters they belong to, is the greatest. Due to a very different approach to determining good clusters and detecting outliers by these two algorithms, the anomaly classification result will also be different for each of the algorithms. We can design the anomaly search process in a qualitative set in two steps. Initially, all algorithms for the low anomaly threshold can search for common anomalies. If the process does not return results, you can increase the threshold and see if there are common outliers in the set this time.

5.3 Detection of common outliers

We should notice the relationship between the number of outliers and the degree of coverage of clustering algorithms in the context of outliers detection. Table 2 presents some interesting results. For each of the analyzed knowledge bases and the three analyzed levels of the number of outliers (5%, 10%, and 15%, respectively), the table presents the number of clusters for each of the algorithms: *ROCK* and *K - modes*, number of outliers detected by each of these algorithms separately, and then the number of common outliers detected by these algorithms and a percentage that these common outliers represent concerning the entire analyzed set. One of the more essential conclusions is that, the more outliers we look for (5%, 10%, or 15%), by running each of the two analyzed algorithms separately, the more common outliers are found by these algorithms. For example, we found 3, 6, and 8 common outliers in the lymphography dataset, respectively, for the 5%, 10%, and 15% outliers we searched. There are also interesting results in the *BM_attack* dataset. In regard to the number of outliers we searched for, the number of actually found outliers and common outliers did not change (2 common outliers no matter how many outliers we were looking for). It is worth looking at the structure of this data set. It contains the fewest attributes and possible values of these attributes when compared to the rest of the sets, which brings about difficulties with regards to distinguishing objects from each other and detecting a greater or lesser number of outliers. In general, when analyzing all sets, one can notice a specific influence the number of attributes and their values have on the efficiency of outlier detection. The more attribute values there are, the greater the coverage of commonly

detected outliers. This is easily explained. With a greater number of features describing the objects, we achieve a greater differentiation, so it is easier for us to correctly (not accidentally) determine the outliers.

Table 2. The results of % of common outliers obtained for 5%, 10%, and 15% of outliers in each of the datasets

Dataset	%	Clusters		Outliers		common outliers	% of common outliers
		ROCK	K-modes	ROCK	K-modes		
lymphography	5%	2	5	5	5	3	0,020000
	10%	3	5	16	11	6	0,040500
	15%	3	5	16	21	8	0,054100
covid	5%	3	6	7	6	1	0,004900
	10%	5	6	34	25	13	0,063700
	15%	5	6	34	25	13	0,063700
SPECT	5%	6	4	8	10	4	0,014980
	10%	1	4	23	21	10	0,037450
	15%	6	4	50	47	31	0,116100
BM_attack	5%	20	3	3	44	2	0,006200
	10%	20	3	3	44	2	0,006200
	15%	20	3	3	44	2	0,006200
primary-tumor	5%	3	5	12	24	8	0,023599
	10%	6	5	40	24	19	0,056000
	15%	6	5	40	53	28	0,082596
wiki	5%	2	9	28	24	13	0,014240
	10%	1	9	78	96	44	0,048193
	15%	5	9	143	143	74	0,081100

5.4 Evaluation of the proposed methods

As part of this work, a vast number of experiments were performed. We changed the values of individual parameters to observe changes in the cluster structure, the number of generated outliers, and most importantly, in assessing whether the analyzed clustering algorithms return similar results in terms of outliers. In the study, we considered real datasets which frequently contain unusual data. They are not the result of a measurement error, but they differ from most data in the set. It is not always the case that one or more objects stand out significantly from the rest, and we can easily see it. Sometimes, it is also the case that specific subsets of objects differ to the same extent from most of the data. The problem becomes even more complicated when we take into account the fact that these objects in the sets may be more or less differentiated by the specificity of the domain they come from, but also by the method of describing these data (the number of attributes, the number of possible values of these attributes, the number of objects). When objects are described on a categorical (qualitative) scale, the effectiveness of their proper clustering and outlier detection is necessary for a deeper study. Hence, in this paper, we analyze selected clustering algorithms which exemplify two types

of clustering: hierarchical (*ROCK*) and non-hierarchical (*K – modes*). Analysis of the results allows us to conclude that if we care about the speed of calculations or have a large dataset, a good choice will be to use the *K – modes* algorithm. The algorithm is recommended to be used in datasets that we know are divided into a small number of large clusters. Then the initially drawn centroids will have less influence on clustering quality. In most cases, the most reasonable approach is to use the *ROCK* method because it performs an exhaustive analysis of the dataset in search of outliers - it approaches object variables individually. It looks for relationships between objects and variables (attributes and their values). The main disadvantage of this algorithm is a very high computational complexity, which in extreme cases may be close to the cube of the number of objects in the set. For this reason, the algorithm is a good choice if we have small datasets, up to 1000 records. Another difficulty is the selection of the distance between the clusters and the minimum number of clusters. The algorithm execution time and clustering quality are improved by knowing an estimated number of clusters in the set and how far the elements should be apart from each other to not be included in a common cluster. Let us suppose that we do not have an exhaustive knowledge about the dataset. In that case, it is worth running the algorithm many times and analyzing the generated clusters to assess the quality of the parameters.

6 Conclusions

This paper focuses on searching for outliers in qualitative data sets depending on the type and the number of variables. Section 3 describes relatively novel approaches to qualitative clustering data. The results presented in this paper are based on six datasets characterized by a different structure. While there is a multitude of solutions related to quantitative data, clustering data containing only qualitative variables remains a challenge for data scientists. The authors attempted to compare the effectiveness of cluster and outlier detection in qualitative datasets, between which there is no explicit comparison so far. Algorithms based on quantitative data generally tend to have better mathematical properties. This does not apply to qualitative sets, so it is difficult to determine which algorithm works better on the data, and it is difficult to detect natural groups. We define the performance of algorithms in terms of their scalability and cluster generation time. We can draw a primary conclusion from the research that the data structure significantly impacts the algorithm's time complexity. The *K – modes* algorithm defines clusters and outliers as objects far away from modes if we have visible modes in a data set. Otherwise, the optimal number of clusters can be very large or very small, and objects that should be in separate clusters will be in one due to a small distance from central modes. Then, it is better to use the *ROCK* algorithm, which is less efficient and has a much greater computation complexity but is not sensitive to unusual data distribution. We should adequately select the algorithm for a dataset. Each algorithm classifies outliers differently and the results will differ. Algorithms based on categorical data clustering are relatively new methods of detecting outliers in data, having no implementation in commonly used programming languages. The discussed *ROCK* and *K-modes* algorithms introduce different methods to solve this problem and give different solutions in terms of their performance concerning the time needed

to execute the algorithms when the number of records and dimensions change. The quality of the created clusters is measured by the user's knowledge and the examination of the results. The user sets basic parameters of clustering, which require an extensive knowledge of the data [1].

References

1. Carletti, M. and Terzi, M. and Susto, G. A.: Interpretable Anomaly Detection with DIFFI: Depth-based Feature Importance for the Isolation Forest. IEEE, US, arXiv preprint arXiv:2007.11117, 1–12 (2000).
2. Breunig, M. M. and Kriegel, H.P. and Ng, R. T. and Sander, J.: LOF: Identifying Density-Based Local Outliers. In: Proceedings of the 2000 ACM SIGMOD international conference on Management of data, 93–104 (2000).
3. Gibson, D. and Kleinberg, J. and Raghavan, P.: Clustering Categorical Data: An Approach Based on Dynamical Systems. In: Proceedings of the 24th International Conference on Very Large Data Bases, The VLDB Journal, 222–236 (2000)
4. Guha, S. and Rastogi, R. and Shim, K.: ROCK: A Robust Clustering Algorithm for Categorical Attributes. In: Information Systems, Elsevier, **25**, 345–366 (2000)
5. Huang, Z.: A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining. In: Proceedings of Data Mining and Knowledge Discovery, 283–304 (1998)
6. Jiang, M. F. and Tseng, S. S. and Su, C. M.: Two-phase clustering process for outliers detection. In: Pattern Recognition Letters, Elsevier, **22**, 691–700 (2001)
7. Kaufman, L. and Rousseeuw, P. J.: Finding Groups in Data: An Introduction to Cluster Analysis. In: John Wiley & Sons, (2005)
8. Liu, F.T. and Ting, K.M. and Zhou, Z.-H.: Isolation Forest. In: IEEE International Conference on Data Mining, 413–422 (2009)
9. Loureiro, A. and Torgo L. and Soares, C.: Outlier detection using clustering methods: a data cleaning application. In: Proceedings of KNet Symposium on Knowledge-Based Systems for the Public Sector, (2004)
10. Primary Tumor, UCI Machine Learning Repository, <https://archive.ics.uci.edu/ml/datasets/Primary+Tumor>. Last accessed 23 May 2020
11. Lymphography, UCI Machine Learning Repository, <https://archive.ics.uci.edu/ml/datasets/Lymphography>. Last accessed 23 May 2020
12. SPECT, UCI Machine Learning Repository, <https://archive.ics.uci.edu/ml/datasets/spect+heart>. Last accessed 9 June 2021
13. COVID, UCI Machine Learning Repository, <https://www.kaggle.com/anushiagrawal/effects-on-personality-due-to-covid19>. Last accessed 9 June 2021
14. Elbow method, [https://en.wikipedia.org/wiki/Elbow_method_\(clustering\)](https://en.wikipedia.org/wiki/Elbow_method_(clustering)). Last accessed 9 June 2021
15. Wiki, UCI Machine Learning Repository, <https://archive.ics.uci.edu/ml/datasets/wiki4HE>. Last accessed 9 June 2021
16. BM_attack, UCI Machine Learning Repository, <https://archive.ics.uci.edu/ml/datasets/Dishonest+Internet+users+Dataset>. Last accessed 9 June 2021
17. Silhouette method, [https://en.wikipedia.org/wiki/Silhouette_\(clustering\)](https://en.wikipedia.org/wiki/Silhouette_(clustering)). Last accessed 9 June 2021