Detecting Outliers in Context of Clustering Imbalanced Categorical Data

Weronika Łazarz
Institute of Computer Science/Faculty of Science and Technology/University of Silesia Katowice, Poland *weronika.lazarz@us.edu.pl*

Agnieszka Nowak-Brzezińska
Institute of Computer Science/Faculty of Science and Technology/University of Silesia Katowice, Poland *agnieszka.nowak-brzezinska@us.edu.pl*

Abstract

Unsupervised models are becoming increasingly common in business processes. They are extremely effective in cases where we don't have a clearly defined decision class or the data contains anomalies that are hard to identify. The problem emerges in the effective processing of categorical data. Recently, many new approaches have been designed to analyze this data type. Still, most of them do not address the issue of imbalanced datasets, which is extremely difficult to catch when dealing with unlabeled data. Moreover, it is sometimes challenging to determine when abnormal observations represent a small data cluster and when they are already anomalies. This research analyzes several less popular algorithms that solve this problem and automatically place abnormal observations into separated clusters. We have shown that such methods are much better at clustering imbalanced data but also perfectly detect outliers in categorical datasets.

Keywords: clustering, qualitative data, outliers, imbalanced data.

1. Introduction

Numerous algorithms are accessible for analyzing numerical data and assessing object similarity through mathematical distance [9, 17, 18, 22]. Conversely, practical scenarios frequently involve qualitative data, encompassing categorical variables, textual elements, or numerical ranges. Qualitative data examples include in-depth interviews, written documentation, interviews, and individual numerical and descriptive data associated with the subject under study.

A good example of the considerable potential of qualitative analysis lies in a medical consultation with a patient who hasn't documented precise health parameter readings daily but can describe them using words or ranges. Qualitative data frequently arise in survey inquiries with multiple-choice responses. To derive insights from such datasets, methods accounting for individual value frequencies are necessary, like histograms or techniques examining value co-occurrences. Moreover, when datasets are labeled, there are many methods of inference. With datasets containing numerous categorical variables, we can accurately forecast decision class behavior using supervised methods. While many such methods exist, they may sometimes prove insufficient, notably when lacking a target variable [35, 19].

We can consider analyzing people infected with a new virus we know little about. Then, grouping the data by patients' health parameters or characteristics indicated in medical interviews allows for the analysis of the entire group, which has similar characteristics. This greatly accelerates the implementation of effective treatment and understanding of the nature of the virus. Qualitative data also appears in many strategies, marketing, or medicine, especially when we describe the system's behavior in words. Regarding health parameters, we prefer "low pressure" or "high temperature" rather than giving an exact value. Since health parameters often change very quickly, it is sometimes better to write them down in words or a range, e.g., body

temperature ranging from 38.2 to 38.7 degrees Celsius. Standard data clustering methods use binary encoding of variables into numbers and then use mathematical measures to discover clusters in the data. This approach possesses several limitations. Firstly, these methods excel at identifying clusters when dealing with continuous data due to tailored measures of similarity and dissimilarity among objects. Secondly, encoding data into binary format results in the loss of valuable information regarding the co-occurrence and frequency of categorical values within the dataset. Thirdly, generating clusters becomes computationally intensive, particularly when handling datasets with numerous categorical variables, leading to a substantial increase in time and memory consumption. Furthermore, a significant benefit of unsupervised qualitative techniques is their inherent capability to segregate outliers into smaller clusters autonomously during the clustering process. In contrast, employing quantitative methods necessitates prior outlier detection and removal before modeling. This task becomes particularly arduous when available anomaly detection methods primarily cater to quantitative rather than qualitative data.

At the data preprocessing stage, the standard task is to make the collection well-balanced. There are many methods to balance a dataset when we have a decision class. The problem arises when the set has no target variable, and the data is categorical. In addition, if we have to process a categorical dataset about whose structure we understand little, it is hard to judge when the small data clusters generated are due to imbalanced data and when they are outliers. Traditional data clustering methods can be disrupted by outliers in the set or with natural small clusters in the data. Developing clustering methods for such cases is very important because, in reality, we often deal with qualitative data that come from new or one-off processes, and the target variable is unavailable. In particular, clustering of qualitative data can be applied in medicine, where readings of health parameters change very quickly, so it is easier to collect data in the form of ranges or words. Detecting outliers in such cases can help diagnose sick patients and implement treatment more quickly. Similarly, detecting small data clusters can identify individuals with unusual but similar symptoms, such as those infected with the same virus.

Therefore, in this research, we decided to test the available methods for clustering qualitative data in terms of how well they separate outliers from the rest of the set and what effect balancing the dataset has on the results. The methods we reviewed are hierarchical or partially iterative. They use standard mathematical measures of the similarity of objects in clusters but evaluate the cluster based on the categories of qualitative variables found in it. Thus, they spontaneously isolate observations dissimilar to the rest in small clusters, both if the collection is imbalanced and these observations form smaller clusters and if they are outliers.

2. Related Work

The predominant focus of research on data clustering algorithms has centered on quantitative data. Distance and partition methods have emerged as prominent contenders in this domain, with notable representatives including the well-established K-means [18] and Agglomerative clustering [22]. Several iterations and adaptations of these methods have been developed over time. However, in response to the escalating demand for text processing capabilities, recent years have witnessed the emergence of numerous novel algorithms explicitly tailored for clustering qualitative data domains. Foremost among these is the K-modes algorithm, an adaptation of the widely utilized K-means algorithm. Renowned for its efficiency and adeptness at identifying clusters characterized by similar attribute values, the K-modes algorithm has garnered widespread adoption [16].

Dedicated algorithms tailored for categorical data include ROCK [12] and CACTUS [11]. These methodologies exhibit superior data clustering capabilities and proficiently identify outliers compared to many conventional algorithms. However, their notably high computational complexity demands robust hardware resources, particularly when handling extensive datasets. Inspired by the dynamic system concept, the ROCK algorithm draws inspiration from the STIRR

method. The algorithm creates subsets of a dataset (hypergraph) to amalgamate akin objects. Also, the CLICKS algorithm uses a graph approach. It partitions data into graphs and extracts maximal cliques from each segment [34]. Maximal cliques represent cohesive groups of data points sharing analogous categorical attributes.

The challenge of time complexity and the influence of outliers on categorical data clustering has been tackled by the creators of the Clope algorithm [33] and its adaptations, such as the scalable version, SClope [23]. Employing a comparable strategy, the Squeezer algorithm constructs histograms of unique categories present in clusters [14]. Like the Clope algorithm, it iterates through dataset objects to associate each with the most compatible cluster regarding shared categorical values. Another algorithm with an iterative methodology is Coolcat [1], which utilizes the probability of values pertaining to a cluster. A recent addition to the array of techniques for clustering qualitative data is the Fair-Multiclustering algorithm [28]. The authors introduce the concept of "fairness constraints" to ensure each cluster meets specific fairness criteria, such as demographic balance or proportional representation.

New work on outlier detection in qualitative data has appeared in the last few years. The available methods deal with data preprocessing, not the modeling process itself. The authors [32] describe several methods for detecting anomalies in qualitative sets based on categories of variables. The paper [15] describes a dynamic approach to analyze communication data extracted from a WebAssembly sandbox to capture application behavior better. The authors of [24] propose a CRBW method for identifying outliers in categorical data with varying frequency distributions.

3. Methods under study

We discuss three types of data clustering algorithms: partitioning algorithms (hierarchical agglomerative), iterative algorithms with a predefined number of clusters, and iterative algorithms governed by an object similarity parameter. The computational complexity of the described algorithms is shown in Table 1. The group of agglomerative hierarchical algorithms includes two of those studied - ROCK and Fair-Multiclustering.

In the case of iterative algorithms, the user defines the number of clusters to achieve. By far, the best-known algorithm of this type is K-means. The main disadvantage of the algorithm is that if there are outliers in the set, they will significantly impact clustering and, therefore, may strongly push the cluster boundary and accept more objects inside than indicated. An alternative to the K-means algorithm is a version for categorical data - the K-modes algorithm and Coolcat.

There are also iterative algorithms, in which the user does not decide how many clusters to achieve but how similar the objects must be to each other to be merged or to the cluster to be found. These methods start with one single-element cluster. They then iterate over the entire dataset and match each object to an existing cluster or create a new one-element cluster, depending on what maximizes the profit function of such an operation (or minimizes the cost). Finally, they perform several iterations over the entire dataset to move the objects that least fit into the clusters they are in. The Clope and Squeezer methods discussed belong to this group.

Table 1. Algorithms for clustering qualitative data used in the study (n-size of dataset, m-length of vector, k-number of clusters, p-number of categories of qualitative variables).

Algorithm	Param	Type	Time complexity
ROCK	θ, k	hierarchical	$O(n^2 \log n)$
$Clope$	r	iterative	$O(nmk)$
Squeezer	θ	iterative	$O(nmk)$
Coolcat	k	iterative	$O(nmk)$
Fair-Multiclustering	k	hierarchical	$O(pn \log n)$

3.1. Robust Clustering Using Links - ROCK

The ROCK algorithm [25] is a hierarchical clustering algorithm for categorical data. The algorithm is based on the concept of neighbors and links. The neighbors of a point are those points that are significantly similar to it. The algorithm uses a similarity function between objects and between clusters. The user defines a threshold for which pairs of points with a similarity function value greater than or equal to this value are considered neighbors. The number of links between pairs of points is the number of common neighbors of those points. The greater the number of links between a pair of points, the more likely they belong to the same cluster. Starting with one-element clusters, the algorithm repeatedly links the two closest clusters until the desired number of clusters remains or a situation arises where no two clusters can be linked. The connection of C_i and C_j clusters is called $link[C_i, C_j]$ and defined in the following form:

$$link[C_i, C_j] = \sum_{x_1 \in C_i, x_2 \in C_j} link(x_1, x_2), \tag{1}$$

where $link(x_1, x_2)$ is the number of points, such that $sim[x_1, x_2] \geq \alpha$ for α selected from the $[0, 1]$ range. A frequently used measure of the $sim[x_1, x_2]$ probability in the solid clustering method is the $Jaccard\ index$. The goodness measure is given by the equation

$$g(i, j) = \frac{link[i, j]}{(n_i + n_j)^h - n_i^h - n_j^h}, \tag{2}$$

where $h = 1 + 2f(\theta)$ with $f(\theta)$ denoting a dataset-specific function that satisfies the property that each object belonging to the C cluster has an approximate number of neighbors equal to n^h and $(n_i + n_j)^h$ is the number of neighbors in the cluster that was created by merging the clusters C_i and C_j.

3.2. Clope

Clope is based on unique values present in the object. The input of the algorithm is not a traditional dataset but a set of vectors of any size that store values identifying the object in any order. The algorithm starts with one single-element cluster consisting of the first object in the dataset. Each successive record is read and added to a new or existing cluster, depending on which operation maximizes the $Profit$ function. The algorithm iterates again through the entire dataset, calculating the $Profit$ of removing a record from its cluster and placing it in another (existing or new cluster). If moving an object to another cluster increases the $Profit$ function, the element will be moved. Iterating over all objects again is performed until at least one element changes its cluster.

To define the $Profit$ function, we need several other definitions. Let t denote the object in the set X, C denote the set of clusters, $W(C_i)$ the number of unique categories in cluster C_i, the function $|\cdot|$ denotes the length of vector or power of the set and $S(C_i) = \sum_{t \in X} |t|$. The $Profit$ function is expressed by the formula

$$Profit(C) = \frac{\sum_{i=1}^{k} \frac{S(C_i)}{W(C_i)^r} * |C_i|}{\sum_{i=1}^{k} |C_i|}, \tag{3}$$

where k is the number of clusters. The r value is a user-selectable parameter. It is a real number and regulates the similarity level of objects in a common cluster.

The basic assumption of the algorithm is the maximization of the $Profit$ function and the fact that the better defined the cluster is, the fewer unique values it contains and the more objects in it identified by these values. An additional advantage is placing objects in one-element clusters if such action maximizes $Profit$, so outliers are separated from the rest.

3.3. Squeezer

The Squeezer algorithm ($Algorithm$ 1) aims to generate compact and well-separated clusters in multidimensional data by updating the cluster centers iteratively while giving more weight to points closer to the current center. The construction of a new cluster is assisted by a user-selected parameter θ. If all cluster similarity values for an element are less than θ, the element is placed in the new cluster. After going through all the elements in the dataset, the algorithm makes a few more passes to move between the clusters of objects that match the least. The similarity of an object to a cluster is expressed by the formula

$$Sim(C,t) = \frac{1}{|C|}\sum_{p\in t}|t_c \in C : p \in t|, \tag{4}$$

where t is object that we want to append to the C cluster, t_c is object belonging to cluster C, p is value of object t. The approach helps mitigate the impact of outliers, leading to improved clustering performance.

Algorithm 1 Squeezer algorithm

$input$: X-dataset, θ-minimum similarity value $output$: A set of k clusters
Initialization Place first element in single-element cluster. For the rest of objects t in the dataset, perform:

1. For each cluster calculate $Sim(C,t)$
2. Select the largest value V of all calculated similarities from previous step
3. If $V > \theta$ then place object in the corresponding cluster. Otherwise place object in a new single-element cluster.

3.4. Coolcat

The Coolcat algorithm ($Algorithm$ 2) is an entropy-based approach for clustering categorical data. It works by iteratively assigning data points to clusters in a way that minimizes the Shannon Entropy within each cluster, where entropy measures the disorder or uncertainty of a set of categorical values. Entropy value is given by formula

$$SE(C) = -\sum_{p\in P}\frac{|p\in C|}{|C|}\log\frac{|p\in C|}{|C|}, \tag{5}$$

where P is set of all unique values in cluster C and $|C|$ denotes power of cluster C.

Algorithm 2 Coolcat algorithm

$input$: X-dataset, k-number of clusters, m - number of objects to be moved in iteration stage $output$: A set of k clusters
Initialization

1. For each pair of objects t_1, t_2 inX calculate Shannon Entropy value
2. Select a pair of records with the highest entropy value. These are the first two clusters
3. Find $k-2$ objects with the highest value of SE(C, t) and place them in single- element clusters.

Iteration

1. For each object t and cluster C where t is located, calculate $P(C,t)$
2. Select m objects with the highest value of $P(C,t)$
3. For each object from previous step, find the cluster that will reduce the entropy value the most and move object there.

At the beginning, it draws points initiating clustering. Then, it iterates through the entire set and matches the elements to the cluster to minimize the entropy value. At the iteration stage, it moves objects from cluster to cluster, maximizing the probability value of an object belonging to a cluster. The probability is expressed by the formula $P(C,t) = \Pi_{p \in t} \frac{|t_c \in C : p \in t|}{|C|}$, where t is object that we want to append to the C cluster, t_c is object belonging to cluster C, p is value of object t. The Coolcat algorithm aims to create clusters with low entropy, indicating high cohesion within clusters and high separation between clusters. Minimizing entropy effectively identifies homogeneous groups of categorical data points while maximizing the distinction between clusters.

3.5. Fair-Multiclustering

Fair-Multiclustering works on the assumption of a protected attribute existing in the dataset, along with specific proportions between its values. The goal is to divide the entire dataset into clusters that are both well-defined in terms of object similarities and fair from the perspective of protected attributes. Homogeneous clusters contain only similar observations, while equitable clusters ensure that the proportions of protected attribute values closely match the desired proportions.

The first step of the algorithm creates as many clusters as there are unique values in the entire dataset. A single cluster is defined by such a single value. In the next step, the single clusters are combined to produce as many clusters as there are all combinations of unique values between attributes. Each resulting cluster will be defined by as many identifiers as there are attributes in the dataset. Such clusters are called $Multiclusters$. In practice, most of these combinations will be empty, that is, no element in the dataset will match a cluster defined in this way. We then combine pairs of clusters with the highest number of matches. For example, the number of convergences between clusters defined by the values of AB and AC, respectively, is one - the value of A is common. The clusters obtained in this method are already optimal, and the last step of the algorithm is to merge the clusters, which will result in the ratio of protected attributes in the cluster being close to the desired one. For example, if in the dataset, we have the variable "eye color" with the values "blue" and "green" in a ratio of 3:2, we expect that the final clusters will have a similar ratio of these values. This method is useful when specific objects' characteristics strongly influence the formation of clusters, while empirically, we can judge that they should not have an influence. A business example might be a model whose goal is to separate fraud customers from the rest. In practice, often the people committing financial fraud are of foreign nationality, but we don't want the model to overfitting such a pattern.

4. Research

Our research aimed to compare methods for clustering qualitative data regarding the quality of the clusters and the right-detected outliers in the data. In particular, we focused on seeing which methods are better candidates for clustering datasets that are poorly balanced against natural clusters in the data or contain anomalies.

In addition to the custom data clustering methods mentioned earlier, we also tested how standard K-means, K-modes and agglomerative hierarchical methods behave. For this purpose, we binary encoded the data into numeric form. These algorithms do not allow to detection of outliers in the data by themselves, so we applied outlier detection methods for each of them - Local Outlier Factor [3], Minimum Covariance Determinant [27], Isolation Forest [17] and DBSCAN [9] - before running clustering.

Table 2. Datasets used in the research

dataset	rows	columns	values	clusters	dataset	rows	columns	values	clusters	dataset	rows	columns	values	clusters
s1	1000	5	10	2	s15	1000	20	100	10	agaricus-lepiota [21]	8124	22	116	2
s2	1000	5	10	5	s16	1000	20	400	2	balance-scale [30]	625	4	20	3
s3	1000	5	10	10	s17	1000	20	400	5	breast-cancer[36]	286	9	41	2
s4	1000	5	25	2	s18	1000	20	400	10	car [2]	1728	6	21	4
s5	1000	5	25	5	s19	1000	100	200	2	flare	323	9	22	2
s6	1000	5	25	10	s20	1000	100	200	5	house-votes[6]	435	16	32	2
s7	1000	5	100	2	s21	1000	100	200	10	promoters [13]	106	57	228	2
s8	1000	5	100	5	s22	1000	100	500	2	solar-flare[31]	1066	11	41	2
s9	1000	5	100	10	s23	1000	100	500	5	spect [5]	267	22	44	2
s10	1000	20	40	2	s24	1000	100	500	10	splice[20]	3190	60	287	3
s11	1000	20	40	5	s25	1000	100	2000	2					
s12	1000	20	40	10	s26	1000	100	2000	5					
s13	1000	20	100	2	s27	1000	100	2000	10					
s14	1000	20	100	5										

4.1. Datasets

In our research, we used real data and artificially generated collections. Real data includes sets available from open online sources (Table 2). These are labeled data with qualitative characteristics. Some of the datasets are well-balanced concerning class, and some retain a large disparity between classes, so we were able to assess what effect the imbalanced dataset has on the data clustering results. Sometimes, small-volume classes can be suspected of being outliers in the data because they contain a very small sample of objects. With the objective variable, we could assess whether the algorithms cluster the data correctly. Before we started clustering, the target variable was removed but kept in memory. Some of the analyzed algorithms allow clustering sets with empty fields, but because of the other methods, we filled in the gaps with the most common value. For outlier detection analysis, we used sets artificially generated with different numbers of variables, classes of qualitative variables, and natural clusters in the data. We generated the collections using the Isotropic Gaussian Mixture Model [26] method, added 2% of the deserving observations to each set, and subjected the sets to discretization to obtain the expected variation in terms of categories of variables.

4.2. Input parameters

Since we had sets with a variable target then we could assume that the algorithms should generate just as many clusters as there are decision classes. On the other hand, if there were more clusters, but they mainly stored objects belonging to one class, the effect was also what we expected. We must also take into account that the more clusters, the better the classes will be distributed in them, and building a large number of clusters is not the goal of the algorithms. Therefore, we tried to choose the parameters so that the number of resulting clusters would be the same or close to the number of original classes or not much more. This was not difficult with the Fair-Multiclustering algorithm because it takes the number of clusters as an input parameter. The ROCK method also seeks to obtain as many clusters as the input parameter k, but the number of clusters is also generated by the second parameter θ. We chose this parameter by trial and error, bearing in mind that the higher the value of the parameter, the lower the required similarity of objects in the cluster. The higher the parameter, the greater the disparity between the sizes of the clusters created. If our goal would be to isolate outlier observations from the rest then the θ parameter should be high. In this case, we aimed to form clusters with counts similar to those of the original classes.

The Coolcat algorithm also takes the number of expected clusters as input. The second parameter is m, corresponding to the number of objects least matching their clusters. These

objects are moved in an iterative step. The value of m must not be too large to avoid completely changing the clusters obtained in the initial step of the algorithm. It also can't be too small because some objects may still be mismatched to the dataset. In our research, we started with 50% of the dataset and, evaluating the results, we increased or decreased this value. The expected number of clusters we introduce in the algorithm can produce one large cluster and several smaller clusters. To avoid this, we chose a strategy of iterating over the range of the k parameter to select the optimal one. The algorithm for finding the optimal k takes the elements of the sequence $K, K-1, K+1, K-2, K+2, \ldots, 2, 2K-2$, and in the absence of a satisfactory solution then iterates over the elements of the sequence $2K-1, 2K, \ldots, K+20$, where K is the number of natural classes in the dataset. We face a similar problem when looking for a parameter for the Clope algorithm - sometimes, we get a very large number of small clusters and a smaller-than-expected number of large clusters. Therefore, we choose a parameter that is on the elbow of the function of the dependence of $Profit$ on the parameter r. The number of clusters always increases as the parameter increases, so $Profit$ also increases, but the function is convex or concave depending on the dataset.

Squeezer is the last algorithm that requires a θ parameter selection strategy. The authors propose that for each pair from a randomly selected subset of objects, the input parameter was the average of $sim(t_i, t_j)$ for each pair of objects t_i, t_j from the set D. We propose an average of $sim(C, t)$ values from the first few tens iterations of the algorithm.

4.3. Cluster evaluation metrics

We assessed performance on several well-known measures of cluster quality: Calinski-Harabasz[4], Davies-Bouldin [7], Rand [25], Dunn [8], Fowlkes-Mallows [10] and Shannon Entropy [29]. Shannon Entropy is interpreted here as a measure of the information stored in a cluster. In this case, we want a single cluster to contain as little information as possible. When evaluating clustering measures empirically, we believe that the comparative measure that best captures the similarity between clustering and natural clustering in the data is Shannon Entropy. For obvious reasons, we cannot compare clustering with labels in data directly - the resulting clusters may have been numbered in a different order than the decision class labels, for example, the two clusters A=[1, 0, 1], B=[0, 1, 0] despite their different numbering are identical. Therefore, we used adjusted versions of the Rand, Fowlkes-Mallows, and Shannon Entropy measures. Let C_1, C_2 be vectors $n \times 1$ representing the cluster numbers assigned to consecutive objects in the set, for the two culling algorithms $A_1 and A_2$, respectively. Let k and l be the number of clusters generated by the A_1 and A_2 algorithm. The confusion matrix $M = (m_{ij})$ of the pair C_1, C_2 is a $k \times l$ matrix whose ij-th element is equal to the number of elements in the intersection of the clusters C_{1i} and C_{2j}, $i \in \{1, ..., k\}, j \in \{1, ..., l\}$ - $m_{ij} = |C_{1i} \cap C_{2j}|$.

Adjusted Rand Index is given by formula

$$R(C_1, C_2) = \frac{\sum_{i=1}^{k}\sum_{j=1}^{l}\binom{m_{ij}}{2} - 2\frac{\sum_{i=1}^{k}\binom{|C_{1i}|}{2}\sum_{j=1}^{l}\binom{|C_{2j}|}{2}}{n(n-1)}}{\frac{1}{2}(\sum_{i=1}^{k}\binom{|C_{1i}|}{2} + \sum_{j=1}^{l}\binom{|C_{2j}|}{2}) - 2\frac{\sum_{i=1}^{k}\binom{|C_{1i}|}{2}\sum_{j=1}^{l}\binom{|C_{2j}|}{2}}{n(n-1)}} \tag{6}$$

Fowlkes Mallows Index Assuming that C_1 is the vector obtained by clustering, and C_2 is the vector of actual clusters (classes) in the dataset, Shannon's measure is given by the formula

$$FM(C_1, C_2) = \frac{(\sum_{i=1}^{k})\sum_{j=1}^{l} m_{ij}^2 - n}{\sqrt{((\sum_{i=1}^{k})(\sum_{j=1}^{l} m_{ij})^2 - n)((\sum_{j=1}^{l})(\sum_{i=1}^{k} m_{ij})^2 - n))}} \tag{7}$$

Shannon Entropy for clusterings is given by formula

$$SE(C_1, C_2) = -\frac{1}{n}\sum_{i=1}^{k} p_i \log p_i, p_i = \frac{\sum_{j=1}^{l} m_{ij}}{|C_1|} \tag{8}$$

The Fowlkes-Mallows measure discriminates against situations where the algorithm has generated more clusters than natural clusters in the dataset, even if each generated cluster is dominated by objects from one cluster. In practice, clustering often classifies objects differently from supervised methods because it does not generate rules, so a slightly larger number of clusters than classes of the target variable is acceptable as long as these clusters are well discovered. Therefore, we evaluate the correctness of clustering mostly based on Shannon Entropy.

4.4. Outlier definition

Since none of the algorithms explicitly define an outlier observation, depending on the specifics of the algorithm, outliers were selected in two ways: as single-element clusters or as observations with the lowest probability of belonging to a cluster. The probability of an object t belonging to a cluster C is expressed by the formula $P(t, C) = \sum_{t_i \in t} \frac{|q \in C : q_i = t_i|}{|C|}$, where q is object belonging to cluster C. We used this approach to detect deviations in clustering using the Fair-Multiclustering and Coolcat methods. These two algorithms start with a predefined number of clusters and aim to reach it, so we could not use the first approach.

The few-element or one-element cluster approach is feasible for the other three methods. The ROCK algorithm hierarchically divides the clusters, so objects that do not match the rest can be separated in one-element clusters. Such elements will not be attached to any cluster at the merge stage. The situation is similar to Clope and Squeezer algorithms, which separate "cluster spoiling" elements during each iteration. Objects that introduce new information (here, new values) into clusters are redundant; if the whole process does not create clusters that could provide them with membership, such objects will end up in separated small clusters, which we will indicate as outlier clusters. Due to the analysis of specific, relatively small datasets, we have assumed that the deviation clusters are of size 1 or 2.

5. Experimental results

When analyzing sets without a decision class, it is often difficult to assess whether they are balanced. For qualitative data, in particular, standard methods fail. There are many methods for detecting outliers in data at the preprocessing stage, but such methods fail when we do not know whether the outliers are in the set or whether the set is very imbalanced. The imbalance of a dataset here refers to the size of decision classes or the size of natural clusters in the data. Sometimes, it is difficult to assess whether objects different from the rest should form a small cluster in the data or whether they are already outliers. Therefore, the algorithms we analyzed workaround this problem by isolating observations belonging to small clusters. Outliers are located in one- or few-element clusters. The rest of the dataset is decomposed according to the natural order. If the set is imbalanced, clusters of different sizes will form.

Performing the research, we collected clustering results for each algorithm and each dataset using pre-selected parameter values. This way, we obtained 50 results for real sets and 135 for artificial sets. We evaluated each result using standard and adjusted measures. In tests aimed at detecting deviations, we evaluated the results by checking how many real outliers the algorithms detected. We used standard metrics for evaluating classification (here, classifying an observation as a deviation or normal observation). When analyzing outliers, we focused mainly on artificially generated sets because, in the case of real sets, verifying whether observations classified as deviations deviate from the rest would be a very difficult task and would require domain knowledge.

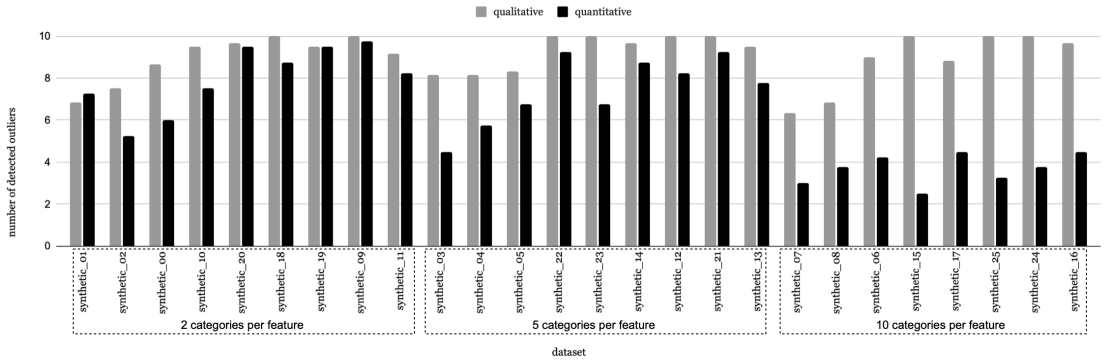

Fig. 1. The number of correctly detected outliers depending on the diversity of the dataset - the average number for qualitative and quantitative algorithms.

Figure 1 shows the averaged result of well-classified objects as outlier observations. This is the average number of correctly detected outliers for two sets of algorithms: qualitative and quantitative. Our observations show that when dealing with qualitative sets of varying values of variables, qualitative methods are much better at detecting deviations in the data. In contrast, commonly used methods in deviation detection, such as LOF or Isolation Forest, do not handle this task well.

Table 3. The value of precision, recall, and f1 classification of outliers for selected datasets.

dataset	CLOPE	ROCK	FAIR-MCLUS	Coolcat	SQUEEZER	IF	LOF	MCD	DBSCAN
PRECISION									
synthetic_3	0.80	1.00	0.60	0.80	1.00	0.30	0.00	0.60	0.83
synthetic_7	0.58	0.47	0.20	0.80	1.00	0.00	0.70	0.00	0.54
synthetic_8	0.60	0.40	0.30	0.70	1.00	0.30	0.70	0.00	0.77
synthetic_15	0.91	1.00	1.00	1.00	1.00	0.00	1.00	0.00	1.00
synthetic_16	1.00	1.00	0.80	1.00	1.00	0.00	1.00	0.30	1.00
synthetic_17	0.77	1.00	0.30	1.00	1.00	0.00	1.00	0.80	0.83
synthetic_24	1.00	1.00	1.00	1.00	1.00	0.00	1.00	0.00	1.00
synthetic_25	1.00	1.00	1.00	1.00	1.00	0.10	1.00	0.00	1.00
RECALL									
synthetic_3	0.80	0.90	0.60	0.80	0.80	0.30	0.00	0.60	1.00
synthetic_7	0.70	0.70	0.20	0.80	0.70	0.00	0.70	0.00	0.70
synthetic_8	0.60	0.80	0.30	0.70	0.70	0.30	0.70	0.00	1.00
synthetic_15	1.00	1.00	1.00	1.00	1.00	0.00	1.00	0.00	1.00
synthetic_16	1.00	1.00	0.80	1.00	1.00	0.00	1.00	0.30	1.00
synthetic_17	1.00	1.00	0.30	1.00	1.00	0.00	1.00	0.80	1.00
synthetic_24	1.00	1.00	1.00	1.00	1.00	0.00	1.00	0.00	1.00
synthetic_25	1.00	1.00	1.00	1.00	1.00	0.10	1.00	0.00	1.00
F1 - SCORE									
synthetic_3	0.80	0.95	0.60	0.80	0.89	0.30	0.00	0.60	0.91
synthetic_7	0.64	0.56	0.20	0.80	0.82	0.00	0.70	0.00	0.61
synthetic_8	0.60	0.53	0.30	0.70	0.82	0.30	0.70	0.00	0.87
synthetic_15	0.95	1.00	1.00	1.00	1.00	0.00	1.00	0.00	1.00
synthetic_16	1.00	1.00	0.80	1.00	1.00	0.00	1.00	0.30	1.00
synthetic_17	0.87	1.00	0.30	1.00	1.00	0.00	1.00	0.80	0.91
synthetic_24	1.00	1.00	1.00	1.00	1.00	0.00	1.00	0.00	1.00
synthetic_25	1.00	1.00	1.00	1.00	1.00	0.10	1.00	0.00	1.00

In Table 3 we have shown the precision, recall and f1 measure values for selected datasets. These are sets with many unique values, for which, in particular, we observed that outliers are better separated by qualitative methods. The two methods that stand out the most from the others are Squeezer and Coolcat (they perfectly detect true outliers). The overall analysis showed that, nevertheless, for datasets with a small number of categorical variables, or when these variables are not differentiated by value, standard methods, particularly the DBSCAN algorithm, perform

better in detecting outliers.

In the context of real data, we researched clustering imbalanced data. Table 4 shows the best results we achieved by clustering the data with the algorithms and the traditional methods: K-means, K-modes and the agglomerative algorithm. The first three metrics indicate the quality of clustering in terms of the content of the final clusters, and the next three indicate the similarity between the natural distribution of objects in the cluster and the distribution obtained after clustering. By natural decomposition, we mean the objects' membership in the decision class. In this study, we show that for imbalanced sets, we achieve the best results using qualitative algorithms. Moreover, knowing the correct distribution of objects in clusters for the studied datasets, we can assess that if we expect to get a lot of data clusters, the Clope and ROCK algorithms (see the car set in Table 4) can best deal with the imbalanced problem. Both algorithms have great potential in clustering qualitative data with unusual structures. In contrast, the Clope algorithm is much faster and allows the exploration of sequential data with non-uniform chain lengths so that we will pay more attention to it in the following studies.

Table 4. The value of the clustering quality measure and the best method for each dataset (ah - agglomerative hierarchical).

		Unsupervised			Supervised		
dataset	**well balanced**	**Dunn**	**Davies -Bouldin**	**Calinski -Harabasz**	**Shannon**	**Rand**	**Fowlkes -Mallwows**
agaricus-lepiota	yes	0,569 (kmodes)	2,239 (ah)	1492 (kmeans)	0,232 (ah)	0,609 (kmodes)	0,815 (kmodes)
balance-scale	no	**0,707 (squeezer)**	**0,998 (coolcat)**	**52 (squeezer)**	**0,457 (coolcat)**	**0,075 (clope)**	**0,654 (coolcat)**
breast-cancer	partially	**0,408 (clope)**	**2,609 (rock)**	40 (kmeans)	**0,594 (clope)**	**0,167 (rock)**	**0,684 (squeezer)**
car	no	**0,548 (clope)**	2,596 (kmeans)	**128 (rock)**	**0,769 (clope)**	**0,119 (rock)**	**0,471 (rock)**
flare	no	**0,56 (coolcat)**	**0,464 (coolcat)**	109 (kmeans)	**0,199 (coolcat)**	**0,200 (clope)**	**0,873 (coolcat)**
house-votes	yes	0,423 (ah)	**1,224 (rock)**	263 (kmeans)	0,341 (kmeans)	0,564 (kmeans)	**0,788 (rock)**
promoters	yes	**0,791 (squeezer)**	3,202 (ah)	3,688 (kmodes)	**0,322 (squeezer)**	0,565 (kmeans)	0,781 (kmeans)
solar-flare	no	**0,387 (clope)**	**2,143 (squeezer)**	219 (kmeans)	**0,153 (rock)**	**0,210 (squeezer)**	**0,933 (clope)**
SPECT	no	**0,556 (coolcat)**	**0,803 (coolcat)**	55 (kmeans)	**0,252 (coolcat)**	**0,152 (fair-mclus)**	**0,816 (coolcat)**
splice	partially	0,570 (ah)	**6,424 (coolcat)**	47 (kmeans)	**0,431 (squeezer)**	0,576 (kmeans)	0,731 (kmeans)

5.1. Algorithms discussion

Let's consider the positive and negative features of the analyzed algorithms. The ROCK algorithm seems to be the best approach to clustering qualitative data because it considers the possible ways to connect clusters. Its significant disadvantage is its high computational complexity. Clope appears to be a competing algorithm, but the $Profit$ function it operates on analyzes only the same classes of features of the object we consider as a member of the cluster and the objects already in it. There is a high probability that natural clustering in the dataset depends mainly on what values occur together on different object features. In such a situation, the Clope method will not generate well-formed clusters. Another disadvantage of this method is the need to store the $Profit$ of each cluster at each stage of the algorithm. Alternatively, we can store its components and calculate the $Profit$ before and after adding an element each time we consider enlarging a cluster. Despite the similar working principle, such a problem does not occur in the Squeezer algorithm. Here, the object's cluster membership function always oscillates within a specific narrow numerical range and is standardized - we divide by the number of elements in

the cluster. Therefore, this value will not increase or decrease as new elements are added to the cluster. This made introducing an input parameter θ possible, which decides whether the object should belong to an existing cluster or form a new one-element cluster. The disadvantage of the algorithm is certainly that it does not perfect the clustering by moving the elements to more matching clusters. Still, it would be difficult to find a stop condition for this approach. The newest of the analyzed algorithms, Fair-Multiclustering, aims to reduce the influence of certain variables on clustering in qualitative data. On the other hand, it requires a specialist to assess whether a variable is subject to fair distribution.

6. Conclusions and Future Work

In this article, we described several unsupervised methods for clustering qualitative data. We have developed ways to self-detect deviations during clustering instead of traditionally before clustering. For qualitative data, there are few effective methods for outlier detection at the data preprocessing stage. Our research shows that when dealing with real qualitative datasets, using qualitative methods to cluster them makes sense. Such methods work well when the data collections are imbalanced or when there are outlier observations. Due to the complex implementation of many algorithms discussed, we chose smaller datasets for analysis. In the future, we plan to expand our research to include large qualitative datasets from real-world processes, particularly amino acid sequence datasets and interviews with choice questions. We evaluated clustering and outlier detection results using standard measures of cluster quality and classification quality. To evaluate the clusters, we had to convert the dataset to binary form. In future work, we plan to build cluster quality measures better suited to qualitative data using Shannon entropy, which is discussed in the methods of detecting outliers in clusters - the probability of an object belonging to a cluster. Another possibility is to evaluate a cluster without analyzing each element separately but considering its unique variables and analyzing di-grams and tri-grams of such values.

References

[1] Barbara, D., Li, Y., and Couto, J.: COOLCAT: An entropy-based algorithm for categorical clustering. In: Nov. 2002, pp. 582–589.

[2] Bohanec, M.: *Car Evaluation*. UCI Machine Learning Repository. 1997.

[3] Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J.: LOF: identifying density-based local outliers. In: *SIGMOD'00 Proceedings*. ACM, 2000, pp. 93–104.

[4] Calinski, T. and JA, H.: A Dendrite Method for Cluster Analysis. In: *Communications in Statistics - Theory and Methods* 3 (Jan. 1974), pp. 1–27.

[5] Cios, K., Kurgan, L., and Goodenday, L.: *SPECT Heart*. UCI Machine Learning Repository. 2001.

[6] *Congressional Voting Records*. UCI Machine Learning Repository. 1987.

[7] Davies, D. and Bouldin, D.: A Cluster Separation Measure. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* PAMI-1 (May 1979), pp. 224–227.

[8] Dunn, J. C.: A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. In: *Journal of Cybernetics* 3.3 (1973), pp. 32–57.

[9] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD'96. 1996, pp. 226–231.

[10] Fowlkes, E. B. and Mallows, C. L.: A Method for Comparing Two Hierarchical Clusterings. In: *Journal of the American Statistical Association* 78.383 (1983), pp. 553–569.

[11] Ganti, V., Gehrke, J., and Ramakrishnan, R.: CACTUS-clustering categorical data using summaries. In: KDD '99. San Diego, California, USA: ACM, 1999, pp. 73–83.

[12] Guha, S., Rastogi, R., and Shim, K.: Rock: A robust clustering algorithm for categorical attributes. In: *Information Systems* 25.5 (2000), pp. 345–366.

[13] Harley, C., Reynolds, R., and Noordewier, M.: *Molecular Biology (Promoter Gene Sequences)*. UCI Machine Learning Repository. 1990.

[14] He, Z., Xu, X., and Deng, S.: Squeezer: An efficient algorithm for clustering categorical data. In: *Journal of Computer Science and Technology* 17 (Sept. 2002), pp. 611–624.

[15] Heinrich, T., Will, N., Obelheiro, R., and Maziero, C.: A Categorical Data Approach for Anomaly Detection in WebAssembly Applications. In: Jan. 2024, pp. 275–284.

[16] Huang, J. Z.: A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining. In: *Workshop on Research Issues on Data Mining and Knowledge Discovery*. 1997.

[17] Liu, F. T., Ting, K., and Zhou, Z.-H.: Isolation Forest. In: Jan. 2009, pp. 413–422.

[18] MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: 1967.

[19] Maji, S. and Arora, S.: Decision Tree Algorithms for Prediction of Heart Disease. In: *Information and Communication Technology for Competitive Strategies*. Ed. by Fong, S., Akashe, S., and Mahalle, P. N. Singapore: Springer Singapore, 2019, pp. 447–454.

[20] *Molecular Biology (Splice-junction Gene Sequences)*. UCI Machine Learning Rep. 1992.

[21] *Mushroom*. UCI Machine Learning Repository. 1987.

[22] Nielsen, F.: Introduction to HPC with MPI for Data Science. Sept. 2016.

[23] Ong, K.-L., Li, W., Ng, W. K., and Lim, E.: SCLOPE: An algorithm for clustering data streams of categorical attributes. In: Jan. 2004, pp. 209–218.

[24] Pang, G., Cao, L., and Chen, L.: Outlier Detection in Complex Categorical Data by Modeling the Feature Value Couplings. In: July 2016.

[25] Rand, W. M.: Objective Criteria for the Evaluation of Clustering Methods. In: *Journal of the American Statistical Association* 66.336 (1971), pp. 846–850.

[26] Reynolds, D.: Gaussian Mixture Models. In: *Encyclopedia of Biometrics* (Jan. 2008).

[27] Rousseeuw, P. and Driessen, K.: A Fast Algorithm for the Minimum Covariance Determinant Estimator. In: *Technometrics* 41 (Aug. 1999), pp. 212–223.

[28] Santos-Mangudo, C. and Heras, A.: A fair-multicluster approach to clustering of categorical data. In: *Central European Journal of Operations Research* 31 (2022), pp. 1–22.

[29] Shannon, C. E.: A mathematical theory of communication. In: *The Bell System Technical Journal* 27.3 (1948), pp. 379–423.

[30] Siegler, R.: *Balance Scale*. UCI Machine Learning Repository. 1994.

[31] *Solar Flare*. UCI Machine Learning Repository. 1989.

[32] Taha, A. and Hadi, A.: Anomaly Detection Methods for Categorical Data: A Review. In: *ACM Computing Surveys* 52 (May 2019), pp. 1–35.

[33] Yang, Y., Guan, X., and You, J.: CLOPE: A Fast and Effective Clustering Algorithm for Transactional Data. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Aug. 2002).

[34] Zaki, M., Peters, M., Assent, I., and Seidl, T.: CLICKS: An effective algorithm for mining subspace clusters in categorical datasets. In: *Data & Knowledge Engineering* 60 (Jan. 2007), pp. 51–70.

[35] Zhang, H., Wang, X., Fu, Z., Luo, M., Zhang, Z., Zhang, K., He, Y., Wan, D., Zhang, L., Wang, J., Yan, X., Han, M., and Chen, Y.: Potential Factors for Prediction of Disease Severity of COVID-19 Patients. In: *medRxiv* (2020).

[36] Zwitter, M. and Soklic, M.: *Breast Cancer*. UCI Machine Learning Repository. 1988.