A matching based clustering algorithm for categorical data

Ruben A. Gevorgyan · Yenok B. Hakobyan

Abstract Cluster analysis is one of the essential tasks in data mining and knowledge discovery. Each type of data poses unique challenges in achieving relatively efficient partitioning of the data into homogeneous groups. While the algorithms for numeric data are relatively well studied in the literature, there are still challenges to address in case of categorical data. The main issue is the unordered structure of categorical data, which makes the implementation of the standard concepts of clustering algorithms difficult. For instance, the assessment of distance between objects, the selection of representatives for categorical data is not as straightforward as for numeric data. Therefore, this paper presents a new framework for partitioning categorical data, which does not use the distance measure as a key concept. The Matching based clustering algorithm is designed based on the similarity matrix and a framework for updating the latter using the feature importance criteria. The experimental results show this algorithm can serve as an alternative to existing ones and can be an efficient knowledge discovery tool.

Keywords categorical data · clustering algorithm · similarity matrix · feature importance

Mathematics Subject Classification (2010) 62H30 · 62H17 · 62H20

1 Introduction

Cluster analysis is one of the “super problems” in data mining. Generally speaking, clustering is partitioning data points into intuitively similar groups (Saxena et al. 2017). This definition is simple and does not consider the challenges that occur while applying cluster analysis to real-world datasets. Nevertheless, this type of analysis is common in different fields, e.g., text mining, marketing research, customer behavior analysis, financial market exploration.

Nowadays various clustering algorithms have been developed in the literature. Each of them has its advantages and disadvantages. Moreover, as the data come in different forms, e.g., text, numeric, categorical, image, the algorithms perform differently in different scenarios. In other words, the performance of a particular clustering algorithm depends on the structure of the data under consideration.

Cluster analysis of numeric data is relatively well studied in the literature. Various approaches are implemented such as representative-based, hierarchical, density-based, graph-based, model-based, grid-based (Sajana et al. 2016). Lately, increasing attention has been paid to partitioning non-numeric types of data. An important topic is the clustering of categorical data. The problem is that the most common clustering algorithms for categorical data are modifications of the ones introduced for numeric data. For instance, K-modes (Huang 1997) is a prototype of the K-means (MacQueen 1967) algorithm. However, several researchers have developed algorithms specifically for categorical data (e.g., Nguyen and Kuol 2019, Chen et al. 2016, Yanto et al. 2016), but there is still much room for new approaches.

The main problem in partitioning categorical data is that the implementation of the standard operations used in clustering algorithms possesses several limitations. For instance, the definition of distance between two objects
with categorical features is not as straightforward as with numeric features, because categorical data takes only discrete values which do not have any order, unlike numeric data. The simplest solution is to transform the categorical data into binary data and then apply one of the common clustering algorithms. Moreover, several novel data transformation approaches have also been developed (e.g. Qian et al. 2015). On the other hand, researchers have developed and employed similarity measures (e.g. Borjia et al. 2008, Gouda and Zaki 2005, Burnaby 1970) to overcome this issue. Another problem is the assessment of cluster representatives because many mathematical operations are not applicable to categorical data. For instance, it is impossible to assess the mean of the categorical feature. Taking into account the limitations of existing algorithms the aim of this paper is to present an algorithm which is not using predefined distance/similarity measures as a key concept and is not based on representatives for assigning data points to clusters. The key concept of the Matching based clustering algorithm (MBC) is that two objects with categorical features are similar only if all the features match. Thus, the algorithm is based on the similarity matrix. In addition, we employ a feature importance framework for choosing which features to drop on each iteration until all objects are clustered. The test on the Soybean disease dataset (Dua et al. 2017) shows that the algorithm is highly accurate and can serve as an efficient data mining tool.

The rest of the paper is organized as follows. We briefly review the common categorical data clustering algorithms in section 2. In section 3 we discuss the categorical data and its limitations. In section 4 we introduce the general framework of the Matching based clustering algorithm. Section 5 presents the experimental results on the Soybean disease dataset. Finally, we summarize and describe our future plans.

2 A review of categorical data clustering methods

Researchers have proposed various methods and algorithms for clustering categorical data. These algorithms can be grouped into five main classes: model-based, partition-based, density-base, hierarchical, projection-based (Berkhin 2006). The main differences between these algorithms are the similarity or distance measures between data points and the criteria which identify the clusters. In the next paragraphs, we discuss the most common approaches in clustering categorical data.

Model-based clustering is based on the notion that data come from a mixture model. The most commonly used models are statistical distributions. This type of algorithms starts with assessing the prior model based on the user-specified parameters. Then the it aims at recovering the latent model by changing the parameters on each iteration. The main disadvantage of this type of clustering is that it requires user-specified parameters. Hence, if the assumptions are false, the results will be inaccurate. At the same time models may oversimplify the actual structure of the data. Another disadvantage of model-based clustering is that it can be slow on large datasets. Some model-based clustering algorithms are SVM clustering (Winters-Merat 2007), BILCOM Empirical Bayesian (Andreopoulos et al. 2006), Autoclass (Cheeseman et al. 1988).

Partition-based clustering algorithms are the most commonly used ones because they are highly efficient in processing large datasets. The main concept is defining representatives of each cluster, allocating objects to the cluster, redefining representatives and reassigning objects based on the dissimilarity measurements. This is repeated until the algorithm converges. The main drawback of this type of algorithms is that they require the number of clusters to be predefined by the user. Another disadvantage is that several algorithms of this type produce locally optimal solutions and are dependent on the structure of the dataset. Several partition-based algorithms are K-modes, Fuzzy K-modes (Ji et al. 2011), Squeezier (He et al. 2002), COOLCAT (Barbára et al. 2002).

Density-based algorithms define clusters as subspaces where the objects are dense and they are separated by subspaces of low density (e.g. Du et al. 2018, Singh and Meshram 2017, Azzalini and Menardi 2016, Gionis et al. 2005). The implementation of density-based algorithms for categorical data is challenging as the values of features are unordered. Even though they can be fast in clustering, they may fail to cluster data with varying density.

Hierarchical algorithms represent the data as a tree of nodes, where each node is a possible grouping of data. There are two possible ways of clustering categorical data using hierarchical algorithms: in an agglomerative (bottom-up) and divisive (top-down) fashion. However, the latter is less common. The main concept of the agglomerative algorithm is using a similarity measure to gradually allocate the objects to the nodes of the tree. The main disadvantage of hierarchical clustering is their slow speed. Another problem is that the clusters may merge, thus these algorithms might lead to information distortion. Several hierarchical clustering algorithms for categorical data are LIMBO (Andritsos et al. 2004), ROCK (Guha et al. 2000), COBWEB (Fisher 1987).

Projected clustering algorithms are based on the fact that in high dimensional datasets clusters are formed based on specific subsets of features. In other words, each cluster is a subspace of high-dimensional datasets defined by a subset of features only relevant to that cluster. The main issue with projected clustering algorithms is that it requires user-specified parameters. If the defined parameters are inaccurate the clustering will be poor. Projected cluster algorithms...
include HIERDENC (Andreopoulos et al. 2007), CLICKS (Zaki and Peters 2005), STIRR (Gibson et al. 2000), CLOPE (Yang et al. 2002), CACTUS (Ganti et al. 1999).

Summarizing the existing algorithms we can conclude that most of them find some trade-off between accuracy and speed. However, considering the growing interest in analyzing categorical data in social, behavior, bio-medical science we are more interested in high accurate algorithms. Furthermore, as one can notice the majority of the algorithms uses some distance, similarity or density metrics and defines representatives of clusters as a subroutine of the algorithms. At the same time, they also require user-specified parameters. These factors can be seen as limitations in case of clustering categorical data. Therefore we propose a new approach to partitioning the categorical data, which avoids these features. To introduce latter, in the next section we discuss the main characteristics of categorical data.

3 Categorical data

Data comes in various forms such as numeric, categorical, mixture, spatial and so on. The analysis of each type of data poses unique challenges. The categorical data is not an exception. This type of data is widely used in political, social and biomedical science. For instance, the measures of attitudes and opinions can be assessed with categorical data. The measures of the performance of medical treatments can also be categorical. Even though the mentioned fields have the largest influence on the development of the methods for categorical data, this type of data also occurs in other fields such as marketing, behavior science, education, psychology, public health, engineering. In this paper, we focus only on categorical features with unordered categories.

For sake of notation, consider a multidimensional dataset $D$ containing $n$ objects. Each object is described by $m$ categorical features each with $k = 1, 2, 3, ..$ unique categories. Thus, the dataset $D$ can be viewed as a matrix below:

$$D_{n,m} = \begin{pmatrix}
  c_{1,1} & c_{1,2} & \cdots & c_{1,m} \\
  c_{2,1} & c_{2,2} & \cdots & c_{2,m} \\
  \vdots & \vdots & \ddots & \vdots \\
  c_{n,1} & c_{n,2} & \cdots & c_{n,m}
\end{pmatrix}$$  (1)

where each object is described by a set of categories $O_i = [c_{i,1}, c_{i,2}, c_{i,3}, \cdots c_{i,m}]$. Also, the frequency of the unique category $c$ in the dataset is defined as $f(c)$.

As the categorical features have discrete values with no order, the application of distance measures such as euclidean distance may yield inaccurate results. However, the most common approach to overcome this limitation is the implementation of data transformation techniques. For instance, one can use binarization to transform the data into binary data and then apply the distance measures. On the other hand, the traditional way of comparing two objects with categorical features is to simply check if the categories coincide. If the categories of all the features under consideration match, the objects can be viewed as similar. This does not mean they are the same, because they can be distinguished by other features. Thus, researchers have proposed various similarity measures instead of requiring all the features to match. The common approach is the overlap (Stanhill and Waltzi (1986)). According to it, the similarity between two objects $O_x = [c_{x,1}, c_{x,2}, \cdots c_{x,m}]$ and $O_y = [c_{y,1}, c_{y,2}, \cdots c_{y,m}]$ is assessed by:

$$Ov(O_x, O_y) = \frac{1}{m} \sum_{i=1}^{m} \gamma_i,$$  (2)

where

$$\gamma_i = \begin{cases} 1 & \text{if } c_{x,i} = c_{y,i} \\ 0 & \text{otherwise} \end{cases}$$  (3)

It can take values from $[0; 1]$. The closer value gets to one, the higher is the similarity between the objects.

While implementing overlap, one can notice that the probability of finding two objects with the same categories rapidly decreases as the number of features and the number of unique categories of each feature increases. The problem is that the overlap measure gives equal weights to the features and doesn’t take into account the importance of each feature in partitioning the data. However, the researchers have proposed more efficient ways of assessing similarity, which take into account the frequency of each category in the dataset. Researchers have introduced various types of similarity measures that are based on this concept (e.g. Jian et al. 2018, dos Santos and Zrati 2015, Alamuri et al. 2014).

Some of them are based on the probabilistic approaches, for instance, Goodall (Goodall 1966):

$$Goodall(O_x, O_y) = \frac{1}{m} \sum_{i=1}^{m} S_i,$$  (4)

where

$$S_i = \begin{cases} 1 - \frac{f(c_{x,i})f(c_{y,i})}{n(n-1)} & \text{if } c_{x,i} = c_{y,i} \\ 0 & \text{otherwise} \end{cases}$$  (5)

Some are based on information-theoretic approaches, for instance, Lin (Lin 1998).

$$Lin(O_x, O_y) = \frac{1}{\sum_{i=1}^{m} (\log(f(c_{x,i})) + \log(f(c_{y,i})))} \sum_{i=1}^{m} S_i,$$  (6)

where

$$S_i = \begin{cases} 2\log\frac{f(c_{x,i})}{n} & \text{if } c_{x,i} = c_{y,i} \\ 2(\log(f(c_{x,i}) + \log(f(c_{y,i})) & \text{otherwise} \end{cases}$$  (7)
Nevertheless, there are still cases when the use of similarity measures can be misleading. For instance, consider the \( D_{4,2} \) dataset with 4 objects and 2 categorical features with \([a_1, a_2], [b_1, b_2]\) categories respectively.

Table 1: An example of data with 4 objects and 2 categorical features

| Object | C   | B   |
|--------|-----|-----|
| \(O_1\) | \(a_1\) | \(b_1\) |
| \(O_2\) | \(a_2\) | \(b_2\) |
| \(O_3\) | \(a_1\) | \(b_2\) |
| \(O_4\) | \(a_2\) | \(b_1\) |

Based on this matrix the corresponding similarities between each unique pair of objects will be:

Table 2: The similarity measure between each unique pairs of objects in \( D_{4,2} \)

| Object | Overlap | Lin | Goodall |
|--------|---------|-----|---------|
| \(O_1, O_2\) | 0.00 | 0.00 | 0.00 |
| \(O_1, O_3\) | 0.50 | 0.50 | 0.42 |
| \(O_1, O_4\) | 0.50 | 0.50 | 0.42 |
| \(O_2, O_3\) | 0.50 | 0.50 | 0.42 |
| \(O_2, O_4\) | 0.50 | 0.50 | 0.42 |
| \(O_3, O_4\) | 0.00 | 0.00 | 0.00 |

From the table, one can notice these measures can be misleading. For instance, one can group \( O_3, O_4 \) to either \( O_1 \) or \( O_2 \) as similarity measures are the same. Therefore, similarity measures are powerful tools, but they should be used with caution. In this regard, one may consider using a quantitative measure to compare the features and choose relatively important ones. Then the objects will be similar if the categories of the selected features match. This is the main motivation of our approach.

Therefore, we employ several feature importance measures. We define the partial grouping power (PGP) of the feature \( l \) in dataset \( D \) as the number of unique matching pairs on the feature divided by the total number of unique matching pairs in the dataset. This is based on the notion that if the feature has relatively higher number of matching pairs than others, it is more likely to group objects. The PGP can be assessed by:

\[
PGP_l = \frac{\sum_{x=1}^{k_l} f(c_x)(f(c_x) - 1)}{\sum_{x=1}^{m} \sum_{j=1}^{k_x} f(c_j)(f(c_j) - 1)} \tag{8}
\]

where \( c_x \) is the unique category of the feature, and \( f(c_j) \) is the frequency of the category in the dataset. This measure takes values from \([0; 1]\). The closer the value to one the higher the importance of the feature in aggregating the objects.

We also define a measure for partitioning power of a feature. We define partial partitioning power (PPP) of the feature \( l \) in dataset \( D \) as the number of unique mismatching pairs on the feature divided by the total number of unique mismatching pairs in the dataset. The PPP can be assessed by:

\[
PPP_l = \frac{n(n-1) - \sum_{x=1}^{k_l} f(c_x)(f(c_x) - 1)}{\sum_{x=1}^{m}(n(n-1) - \sum_{j=1}^{k_x} f(c_j)(f(c_j) - 1))} \tag{9}
\]

This measure takes values from \([0; 1]\). The closer the value to one the higher the importance of the feature in partitioning the objects. Both methods can be used in the analysis. However, one of the measures can be more accurate than the other one depending on the data under consideration because the structure of the data may vary.

We also present another measure for assessing the feature importance. This one is based on the similarity matrix. Similarity matrix is defined as the matrix below:

\[
SM_{n,n} = \begin{pmatrix}
    m_{1,1} & m_{1,2} & \cdots & m_{1,m} \\
    m_{2,1} & m_{2,2} & \cdots & m_{2,m} \\
    \vdots & \vdots & \ddots & \vdots \\
    m_{n,1} & m_{n,2} & \cdots & m_{n,m}
\end{pmatrix} \tag{10}
\]

where \( m_{i,j} \) is a similarity measure between object \( i \) and \( j \) such as Overlap, Lin, Goodall. Through out this paper we will use the count of matches (CM) between two objects as a similarity measure:

\[
CM(O_x, O_y) = \sum_{i=1}^{m} \gamma_i \tag{11}
\]

where

\[
\gamma_i = \begin{cases}
    1 & \text{if } c_{x,i} = c_{y,i} \\
    0 & \text{otherwise}
\end{cases} \tag{12}
\]

The similarity matrix is symmetrical, thus only upper triangular matrix is used in the calculations. Furthermore, the diagonal will also be ignored. Based on the similarity matrix we define the general influence matrix (GIM) as:

\[
IM_{n,n} = \begin{pmatrix}
    I_{1,1} & I_{1,2} & \cdots & I_{1,m} \\
    I_{2,1} & I_{2,2} & \cdots & I_{2,m} \\
    \vdots & \vdots & \ddots & \vdots \\
    I_{n,1} & I_{n,2} & \cdots & I_{n,m}
\end{pmatrix} \tag{13}
\]

where

\[
I_{i,j} = \begin{cases}
    1 & \text{if } m_{i,j} > \alpha \\
    0 & \text{otherwise}
\end{cases} \tag{14}
\]
where $\alpha$ is a threshold, which is bounded by the values similarity measure can take. In this paper, we set $\alpha$ to 0. After the construction, the features or the subset of features under consideration are dropped, and the influence matrix is updated. The matrix after the drop is defined as the partial influence matrix (PIM$_i$) of corresponding feature or subset of features $l$. In this case, the partial grouping power (PGP$_l$) of the feature or subset of features $l$, is assessed by dividing the count of the ones in the PIM$_i$ by the count of ones in the GIM.

$$PGP_l = \frac{\eta_{PIM_i}}{\eta_{GIM}}$$

where $\eta_{PIM_i}$ is the count of ones in the PIM$_i$ and $\eta_{GIM}$ is the count of ones in the GIM.

One can notice that these measures of feature importance depend only on the number of unique matches in the dataset, and the number of categories of each feature does not influence them. In the next section, we present the Matching based clustering (MBC) algorithm, which combines the importance measures of the features and the similarity matrix to partition categorical data into homogeneous groups.

4 Matching based clustering algorithm

Similar to any clustering algorithm the main objective of MBC is partitioning the data into relatively similar groups. The algorithm is defined for categorical data only. However, one can modify it to employ for other types also, but this is out of the scope of this paper. The main idea is, while there are still objects without clusters, the algorithm will choose features to drop based on their importance. Then it will update the similarity matrix and try to cluster the objects based on the new SM. It uses the similarity matrix where the similarity measure between two objects is defined by the CM.

We also use either PGP$_l$ or PPI$_l$ measure to choose the features to drop on each iteration. For the sake of notations, we define $\theta_p$ as the count of the remaining features on iteration $p$. The initial value of $\theta_0 = m$. We consider two objects to belong to the same cluster if $m_{i,j} = \theta_p$. In other words, they are grouped if their categories coincide for all the remaining features on iteration $p$.

The algorithm consists of the following steps:

1. Construct the similarity matrix $SM_{m,n}$.
2. Calculate the $PGP_l$ of each feature.
3. Allocate the objects to clusters based on the similarity matrix. In other words, group two objects ($i$ and $j$), if $m_{i,j} = \theta_p$. If one of the objects is already allocated to a cluster, assign the second one to the same cluster.
4. Check if there are still objects not assigned to any cluster, if yes continue to next step, otherwise terminate.
5. Remove the features with the lowest $PGP_l$. If there are several features, one may consider either dropping all of them or using the $PGP_{l_2}$ to choose which one to drop.
6. Update the similarity matrix.
7. Additionally update the SM using the statement: $\forall$ existing clusters $i$ and $j$, if $m_{i,j} = \theta_p$, then the values of the rows and columns $i$ and $j$, which are equal to $\theta_p$, are set to zero.
8. Return to step 3.

The algorithm stops if all the objects are clustered or the importance of remaining features is the same. As one can notice the algorithm can use also the $PPPI_l$ as feature importance measure. In this case, the features with the lowest values should be dropped. Moreover, step 7 is optional. The main purpose of this step is to avoid the merging of existing clusters, if one avoids this step, the algorithm builds the dendrogram of the data, where each level of the tree corresponds to the clusters after each feature drop.

| Objects | A  | B  | C  | D  | E  |
|---------|----|----|----|----|----|
| O\_1    | a\_2| b\_1| c\_2| d\_1| e\_2|
| O\_2    | a\_2| b\_1| c\_2| d\_3| e\_2|
| O\_3    | a\_2| b\_1| c\_2| d\_3| e\_1|
| O\_4    | a\_2| b\_1| c\_2| d\_3| e\_4|
| O\_5    | a\_1| b\_2| c\_4| d\_2| e\_3|
| O\_6    | a\_1| b\_2| c\_3| d\_4| e\_4|
| O\_7    | a\_1| b\_2| c\_4| d\_2| e\_2|
| O\_8    | a\_1| b\_2| c\_3| d\_4| e\_1|
| O\_9    | a\_1| b\_2| c\_1| d\_1| e\_3|
| O\_10   | a\_1| b\_2| c\_4| d\_2| e\_2|

Fig. 1: An example of the dendrogram build by MBC algorithm

To illustrate how the algorithm works, we will apply it to a simulated dataset A.

Table 3: Simulated dataset A
In this dataset 10 objects are defined by 4 categorical features \( A, B, C, D, E \) with \([a_1, a_2, b_1, b_2, c_1, c_2, c_3, c_4, d_1, d_2, d_3, d_4]\) and \([e_1, e_2, e_3, e_4]\) unique categories respectively.

We initialize the algorithm by constructing the similarity matrix:

\[
S_{10,10} = \begin{pmatrix}
-5 & 4 & 4 & 0 & 0 & 1 & 0 & 0 & 1 \\
-4 & 4 & 0 & 0 & 1 & 0 & 1 \\
-4 & 0 & 0 & 0 & 1 & 0 & 0 \\
-0 & 1 & 0 & 0 & 0 & 0 & 0 \\
-2 & 4 & 2 & 3 & 4 & - & - \\
-2 & 4 & 2 & 2 & - & - \\
-2 & 2 & 5 & - & - \\
-2 & 2 & 2 & - & - \\
-2 & - & - & - & -
\end{pmatrix}
\]

(16)

Then, the importance of each feature is assessed. In this example, we will use the \( PGP_i \) measure. For instance the \( PGP_A \) will be:

\[
PGP_A = \frac{21}{21 + 21 + 10 + 10 + 9} = 0.30
\]

Respectively \( PGP_B = 0.30, PGP_C = 0.14, PGP_D = 0.14 \) and \( PGP_E = 0.13 \). Then as the \( \theta_0 = m = 5 \), all the objects \( i, j \) with \( m_{ij} = 5 \) are grouped. As we can see we have two clusters \([O_1, O_2]\) and \([O_7, O_{10}]\). As we still have some objects left without cluster, we continue to next step. In particular, as the feature \( E \) has the lowest \( PGP \), we drop it and update the similarity matrix. Also to avoid the merging of already existing clusters, we additionally update \( SM \) according to step 7. As similarity between clusters \([O_1, O_2]\) and \([O_7, O_{10}]\) is not equal to \( \theta_1 = 4 \), we will not make additional changes, and the new data view and corresponding similarity matrix will be:

\[
S_{8,8} = \begin{pmatrix}
-4 & 4 & 0 & 0 & 0 & 0 & 0 \\
-4 & 0 & 0 & 0 & 0 & 0 \\
-0 & 0 & 0 & 0 & 0 & 0 \\
-2 & 4 & 2 & 2 & - & - \\
-2 & 4 & 2 & - & - \\
-2 & 2 & - & - \\
-2 & - & - & - & -
\end{pmatrix}
\]

(18)

As \( \theta_1 = 4 \), we will have \([O_1, O_2, O_3, O_4], [O_5, O_7, O_{10}]\) and \([O_6, O_8]\) clusters. However, we still have one more object to assign to a cluster, thus we drop \( C \) and \( D \) and update the similarity matrix.

\[
\begin{array}{cccc}
\text{Objects} & A & B \\
\hline
(O_1, O_2) & a_1 & b_0 \\
O_3 & a_1 & b_0 \\
O_4 & a_1 & b_0 \\
O_5 & a_0 & b_1 \\
O_6 & a_0 & b_1 \\
(O_7, O_{10}) & a_0 & b_1 \\
O_8 & a_0 & b_1 \\
O_9 & a_0 & b_1 \\
\end{array}
\]

\[
S_{4,4} = \begin{pmatrix}
-0 & 0 & 0 \\
-2 & 2 & 2 \\
-2 & - & - \\
\end{pmatrix}
\]

(19)

Fig. 3: The data view and corresponding \( SM \) after the second iteration

But we also check the statement of step 7 between any pair of the existing clusters. As \( \theta_2 = 2 \), the statement is true in the case of \([O_5, O_7, O_{10}]\) and \([O_6, O_8]\). Thus, the values of corresponding rows (1,3) and columns (1,3), which are equal \( \theta_2 = 2 \), are set to zero. The purpose of this modification is that as we are dropping features with low grouping power, the clusters are more likely to merge. Therefore, we may lose important local partitioning of data points. Thus, the final updated similarity matrix will be:

\[
S_{4,4} = \begin{pmatrix}
-0 & 0 & 0 \\
-2 & 2 & 2 \\
-2 & - & - \\
\end{pmatrix}
\]

(19)

However, the second iteration does not group the \( O_9 \). At the same time as the importance of the remaining features \( A \) and \( B \) is the same the algorithm terminates and the object \( O_9 \) forms the fourth cluster. Thus, the final result of the MBC clustering is:

\[
\begin{array}{cccccc}
\text{Objects} & A & B & C & D & E & \text{Cluster} \\
\hline
O_1 & a_1 & b_0 & c_1 & d_2 & e_2 & 1 \\
O_2 & a_1 & b_0 & c_1 & d_2 & e_2 & 1 \\
O_3 & a_1 & b_0 & c_1 & d_2 & e_2 & 1 \\
O_4 & a_1 & b_0 & c_1 & d_2 & e_2 & 1 \\
O_5 & a_0 & b_1 & c_3 & d_1 & e_4 & 3 \\
O_6 & a_0 & b_1 & c_2 & d_3 & e_4 & 3 \\
O_7 & a_0 & b_1 & c_3 & d_1 & e_4 & 3 \\
O_8 & a_0 & b_1 & c_2 & d_3 & e_4 & 3 \\
O_9 & a_0 & b_1 & c_3 & d_1 & e_4 & 3 \\
\end{array}
\]

Table 4: The final clusters for dataset A

The algorithm has some unique characteristics worth mentioning. First, to achieve better performance one can notice that all the changes required in each step should be
done only on the similarity matrix and there is no need to update the dataset. Second, there is no need for user-defined parameters. However, one may specify the number of clusters. In this case, step 7 is ignored and after the construction of the dendrogram, the clusters are formed. The algorithm creates a tree where each leaf is a possible cluster. In the case of the simulated dataset A, the dendrogram will be:

![Dendrogram](image)

Fig. 4: The dendrogram of clusters of the simulated dataset A

Forth, as the algorithm is based on either feature grouping or partitioning power, this information can be used to understand the structure of the data. For instance, this algorithm can serve as a subroutine for feature selection for other clustering algorithms.

5 Experimental results

We have employed MBC to the Soybean disease dataset (Dua et al. 2017) to test its performance on the real-world dataset. It is one of the standard test data sets used in the machine learning community and has often been used to test conceptual clustering algorithms for categorical data. The Soybean data set has 47 observations, each being described by 35 features. Each observation is identified by one of the 4 diseases – Diaporthe Stem Canker, Charcoal Rot, Rhizoctonia Root Rot, and Phytophthora Rot. These are used as indicators of the accuracy of the algorithm.

After applying the MBC to the Soybean disease dataset, we got 18 different clusters.

Table 5: The distribution of clusters by 4 types of diseases

| DT | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|
| D1 | - | 3 | 2 | - | - | 2 | 2 | - | - | - | - | - | - | - | - | - | 1 |
| D2 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| D3 | - | - | - | - | - | - | - | - | 5 | 2 | - | - | - | - | - | - | - | - |
| D4 | - | - | - | - | 3 | - | - | - | - | - | - | - | - | - | - | - | - | - |

As we can see from the table above all the clusters except for one entirely belong to one of the groups mentioned above. In other words, we have only one possible misclassification. However, as already mentioned one may require a specific number of clusters. In this case, one can use the constructed dendrogram based on the MBC algorithm.

![Dendrogram](image)

Fig. 5: The dendrogram for the Soybean disease dataset

If we compare the MBC with K-modes (Huang 1997), the main differences are: K-modes depends on the data order and the initial cluster representatives, it also requires the number of clusters as an input parameter. In the case of MBC, we do not have these limitations.

Conclusion

Data clustering is broadly studied and used in various applications. The best practice approaches are limited to numeric data, but lately, increasing attention has been paid to clustering other types of data, in particular, categorical data. Thus, specific models are being developed for categorical data. The vital issue in clustering categorical data is the notion of the distance or similarity between the observations. Hence, this paper introduces the Matching based clustering algorithm which can serve as an alternative to existing ones. It is based on the main characteristics of categorical data and presents a new framework for clustering categorical data, which is not based on the distance measure. The main concept of the algorithm is the assessment of the similarity matrix, updating the latter based on the importance criteria of each feature or subset of features and grouping only if the categories of objects entirely match. These operations allow clustering categorical data without transformation. Another advantage is the description of the features, as the algorithm allows to identify the ones which cause the partitioning of the data. These can be important in interpreting clustering results. Finally, MBC does not require any initial parameter.

Our future work plan is to develop and implement a modification of the algorithm to cluster mixture data. Furthermore, overcome its limitation and adapt it to clustering big datasets. Such an algorithm is required in a number of data mining applications, such as partitioning large sets of objects into a number of smaller and more manageable subsets that can be more easily modeled and analyzed.

References

Alamuri M, Surampudi BR, Negi A (2014) A survey of distance/similarity measures for categorical data. In:
Qian Y, Li F, Liang J, Liu B, Dang C (2016) Space structure and clustering of categorical data. IEEE Transactions on Neural Networks and Learning Systems 27(10):2047–2059, DOI 10.1109/TNNLS.2015.2451151

Sajana T, Rani CMS, Narayana KV (2016) A survey on clustering techniques for big data mining. Indian Journal of Science and Technology 9(3)

dos Santos TR, Zrate LE (2015) Categorical data clustering: What similarity measure to recommend? Expert Systems with Applications 42(3):1247 – 1260, DOI https://doi.org/10.1016/j.eswa.2014.09.012

Saxena A, Prasad M, Gupta A, Bharil N, Patel OP, Tiwari A, Er MJ, Ding W, Lin CT (2017) A review of clustering techniques and developments. Neurocomputing 267:664 – 681, DOI https://doi.org/10.1016/j.neucom.2017.06.053

Singh P, Meshram PA (2017) Survey of density based clustering algorithms and its variants. In: 2017 International Conference on Inventive Computing and Informatics (ICICI), pp 920–926, DOI 10.1109/ICICI.2017.8365272

Stanfill C, Waltz D (1986) Toward memory-based reasoning. Commun ACM 29(12):1213–1228, DOI 10.1145/7902.7906

Winters-Hilt S, Merat S (2007) Svm clustering. BMC bioinformatics 8 Suppl 7:S18, DOI 10.1186/1471-2105-8-S7-S18

Yang Y, Guan X, You J (2002) Clope: A fast and effective clustering algorithm for transactional data. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining DOI 10.1145/775047.775149

Yanto ITR, Ismail MA, Herawan T (2016) A modified fuzzy k-partition based on indiscernibility relation for categorical data clustering. Engineering Applications of Artificial Intelligence 53:41 – 52, DOI https://doi.org/10.1016/j.engappai.2016.01.026

Zaki MJ, Peters M (2005) CLICKS: mining subspace clusters in categorical data via k-partite maximal cliques. In: Proceedings of the 21st International Conference on Data Engineering, ICDE 2005, 5-8 April 2005, Tokyo, Japan, pp 355–356, DOI 10.1109/ICDE.2005.33