DyClee-C: a clustering algorithm for categorical data based diagnosis

Tom Obry, Louise Travé-Massuyès, Audine Subias

DyClee-C: a clustering algorithm for categorical data based diagnosis

Tom Obry[1,2] and **Louise Travé-Massuyès**[1] and **Audine Subias** [1]
[1]LAAS-CNRS, Université de Toulouse, CNRS, INSA, Toulouse, France
e-mail: {tobry,louise,subias}@laas.fr
[2]ACTIA, 5 Rue Jorge Semprun, 31432 Toulouse
e-mail: tom.obry@actia.fr

Abstract

In data-based diagnostic applications, large amounts of data are often available but the data remains unlabelled because labelling would require too much time and imply prohibitive costs. The different situations, e.g. normal or faulty, must hence be learned from the data.

Clustering methods, also qualified as unsupervised classification methods, can then be used to create groups of samples according to some similarity criterion. The different groups can supposedly be associated to different situations. Numerous algorithms have been developed in recent years for clustering numeric data but these methods are not applicable to categorical data. However, in many application domains, categorical features are key to properly describe the different situations. This paper presents DyClee-C, an extension of the numeric feature based DyClee algorithm to categorical data. DyClee-C is applied to two data sets: a soybean data set to diagnose the disease soybean plants and a breast cancer data set to assess the current diagnosis in terms of recurrence events and prognose possible relapse.

1 Introduction

In the digital age, the amount of data that is recorded by organizations and companies is enormous. If this data is to have added value, it must be possible to extract relevant information automatically. This is why data mining methods appear to be crucial. Among them, clustering methods have an essential role to play. Indeed, data often remains unlabelled because labelling would require too much time and imply prohibitive costs. In diagnostic applications, the different situations, e.g. normal or faulty, must hence be learned from the data. Clustering methods, also qualified as unsupervised classification methods, can then be used to create groups of samples according to some similarity criterion. The different groups can supposedly be associated to different situations.

In the field of clustering, many unsupervised learning algorithms exist. Among the most well-known, we find K-Means [1], [2], DBSCAN [3] and hierarchical ascending classification (HAC) [4]. These algorithms use a numeric distance criterion such as the classical Euclidean distance or the Manhattan distance for high dimensional data sets to determine sample similarity and closest clusters. For this reason, most algorithms are not applicable to categorical data, that is nominal or ordinal qualitative data. However, categorical features are key to properly describe the different situations in many application domains.

As a matter of fact, the metrics quoted above do not make possible to calculate a distance between two samples described by categorical features. Some algorithms have been developed to overcome this problem, such as K-modes [5], ROCK [6] or SQUEEZER [7]. All these algorithms use their own notion of similarity to create clusters.

In this paper, an extension of the DyClee numeric clustering algorithm is proposed [8], [9], [10]. The method, named DyClee-C, applies to categorical data. It must be considered as a building block of the mixed numeric/categorical DyClee version that is under construction. After the presentation of DyClee-C, DyClee-C is applied to two data sets: a soybean data set to diagnose the disease soybean plants and a breast cancer data set to assess the current diagnosis in terms of recurrence events and prognose possible relapse.

The paper is organized as follows. Section 2 presents the basic principles and the different steps of the DyClee algorithm. The categorical extension DyClee-C is introduced in section 3. The tests on the soybean and breast cancer data sets are presented in section 4. Finally, section 5 provides conclusions and perspectives of this work [1].

2 DyClee algorithm

The **D**ynamic **C**lustering algorithm for tracking **E**volving **E**nvironments (DyClee) is an unsupervised learning method. DyClee implements a distance and density based algorithm that features several properties like handling non convex, multi-density clustering with outlier rejection, and it achieves full dynamicity. All these properties are not generally found together in the same algorithm and DyClee hence pushes forward the state of the art in this respect. The first step qualified as *distance-based step* operates at the rate of the data stream and creates micro-clusters putting together data samples that are close in the sense of the L1-norm. Micro-clusters are stored in the form of summarized representations including statistical and temporal information. The second step qualified as *density-based step* uses a density-based approach applied to the micro-clusters to create the final clusters. A cluster is defined as a set of connected micro-clusters where every inside micro-cluster

[1] This work is part of the CIFRE PhD project n^o2017/0047 in collaboration with the company ACTIA. It is also related to the French Occitanie Region project Evolearn.

presents high density and every boundary micro-cluster exhibits either medium or high density. Figure 1 illustrates how the DyClee algorithm works.

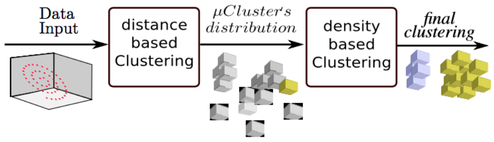

Figure 1: Overview of DyClee

DyClee only handles numeric data sets making use of the KD-Tree algorithm for grouping the micro-cluster at the begining of the density-based step. However, the KD-tree does not process categorical data. A KD-Tree is a space-partitioning data structure for organizing points in a k-dimensional space [11]. KD-Trees are a useful data structure for several applications, such as nearest neighbor search. The work presented in this paper aims to propose a dynamic clustering approach able to manage categorical data and capture large data dimensions. For this purpose, an alternative to the KD-Tree is proposed and integrated with the DyClee method, thus leading to DyClee-C.

3 Categorical extension DyClee-C

The extension to categorical data relies mainly on the removal of micro-clusters and on an alternative solution to KD-Tree for finding neighbors. Indeed, in DyClee, the hypercubes (i.e the micro-clusters) make it possible to represent a similar population. Centers of micro-clusters (μC) can be calculated by averaging, for each attribute, all the samples present in the μC. In the case of categorical data, we decided to experiment an algorithm that does not make use of the micro-cluster concept. In consequence, the distance-based step of DyClee, during which the samples are assigned to the μCs, is removed and the density-based step directly deals with the samples. In DyClee, clusters are formed by searching in each group found by the KD-Tree, sub-regions of dense and semi-dense micro-clusters. In the DyClee-C extension, they are formed from the results of the Locality Sensitive Hashing (LSH) [12] algorithm, [13], [14]. This method is used in several applications such as clustering, searching for nearest neighbors in large dimensions and the detection of similar images.

The following sections detail the principles of the LSH algorithm, how the clusters are formed and they present several parameters of the method allowing to refine the clustering results.

3.1 Locality Sensitive Hashing algorithm

The LSH algorithm is an appropriate alternative to the KD-Tree since, on the one hand, this algorithm is able to process numerical and categorical data, on the other hand this algorithm is a solution to the problem of the *curse of dimensionality* [15], [16], [17]. LSH makes it possible to reduce the dimensionality of large data sets. LSH refers to a family of hash functions that associate samples in *buckets*: similar samples (in the sense of a matching measure of similarity) are assigned to the same bucket while dissimilar samples are assigned to different bucket. Figure 2 illustrates this method: classic hash functions assign each sample to a different bucket while the hash functions belonging to the family of hash functions of the LSH assign close samples to the same bucket.

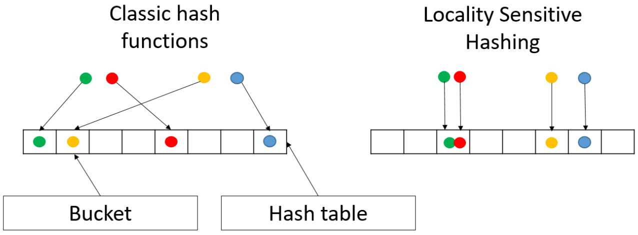

Figure 2: Assignment of samples with hash functions with the LSH

The intuition behind this family of hash functions is that the probability that two samples share the same bucket is related to the distance that separates these two samples. The greater the distance, the lower the probability that the two samples share the same bucket [12]. Definition 3.1 gives the conditions for two samples to share the same bucket.

Definition 3.1. A family of hash functions is called $(d_1, d_2, p_1, p_2) - sensitive$ for all x and $y \in S$, where S is a set of samples if the two following conditions are fulfilled:

- d(x,y) $\leq d_1 \Rightarrow \mathbf{Pr}_{h \in \mathcal{H}}[h(x) = h(y)] \geq p_1$
- d(x,y) $\geq d_2 \Rightarrow \mathbf{Pr}_{h \in \mathcal{H}}[h(x) = h(y)] \leq p_2$

where $d(.,.)$ is the distance between two samples with values between 0 and 1, $h(.)$ is the result of the hash function h applied to a sample. p_1 and p_2 are probability thresholds for two samples to share the same bucket, with values between 0 and 1.
To find sample's neighbors, all samples that share the same basket as some sample at least once on the t hash tables are considered neighbors. This technique is used to find groups in DyClee-C. Figure 3 illustrates the search for nearest neighbors for a sample. The neighbors of the sample q (noted q for *query point*) are all samples that share the same bucket as the q point at least once. Here, the neighbors of q are the red and yellow dots.

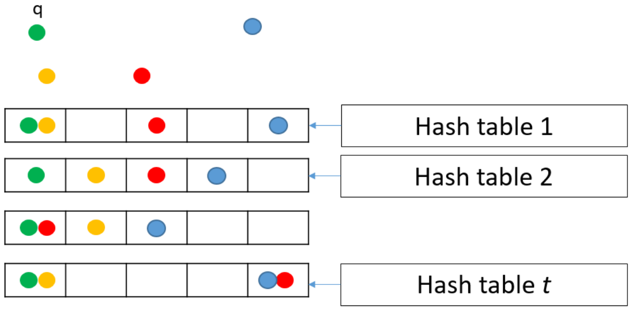

Figure 3: Searching neighbors with the LSH algorithm

3.2 Identification of clusters with DyClee-C

In DyClee, clusters are created from groups of micro-clusters found by the KD-Tree. Let M = {μC_1, ..., μC_z,

..., μC_k }, the set of micro-clusters found with μC_z the z-th micro-cluster and k the number of micro-clusters. D = {D_1, ..., D_z, ..., D_k } is the set of densities corresponding to micro-clusters with D_z the density of the z-th micro-cluster. G = {G_1, ..., G_z, ..., G_l} is the set of micro-cluster groups detected by the KD-Tree, G_z is the z-th micro-cluster group and l is the number of micro-cluster groups. Note that the number of micro-clusters k is generally much bigger than the number of groups of micro-clusters l. For each group, the densest micro-cluster subregions are searched. A cluster is created if a micro-cluster is dense and if its neighbors are dense or semi-dense inside a group. If a micro-cluster is outlier, all the samples in this micro-cluster are considered noise.

DyClee offers two approaches to find clusters: the global approach and the local approach.

In the **global approach**, a micro-cluster μC_z is said to be dense if D_z is greater than or equal to the two global density thresholds which are the median and average density of all micro-clusters D. A micro-cluster μC_z is semi-dense if D_z is greater than or equal to one of the two global density thresholds. Finally, a micro-cluster μC_z is said to be outlier if D_z is less than or equal to the two global density thresholds. These conditions are represented by the inequalities (1), (2), (3), where $median(D_1, ..., D_k)$ and $average(D_1, ..., D_k)$ correspond respectively to the median and average density of micro-clusters in M.

$$\begin{aligned} \mu C_z dense \Leftrightarrow D_z \geq median(D_1, ..., D_k) \\ \wedge D_z \geq average(D_1, ..., D_k) \end{aligned} \tag{1}$$

$$\begin{aligned} \mu C_z semi-dense \Leftrightarrow D_z \geq median(D_1, ..., D_k) \\ \vee D_z \geq average(D_1, ..., D_k) \end{aligned} \tag{2}$$

$$\begin{aligned} \mu C_z outlier \Leftrightarrow D_z < median(D_1, ..., D_k) \\ \wedge D_z < average(D_1, ..., D_k) \end{aligned} \tag{3}$$

In the case of DyClee-C, as explained previously, the concept of micro-cluster no longer exists. The notion of density of a micro-cluster D_z is replaced by the number of neighbors of each categorial sample. The neighbor of a sample is spotted by a *connection*. Let O = {O_1, ..., O_z, ..., O_n }, the set of samples with O_z, the z-th sample of O and n the number of samples. C = {C_{O1}, ..., C_{Oz}, ..., C_{On} } is the set of connections of all samples with C_{Oz}, the number of connections of O_z. C_G = {C_{G1}, ..., C_{Gz}, ..., C_{Gl} } is the set of samples connections for all groups of samples found by the LSH algorithm, with C_{Gz} the number of connections of samples in the group z and l the number of groups of categorical samples found.
As part of the categorical global approach, a sample O_z is dense if C_{Oz} is greater than or equal to the global density thresholds which are then the median and average of the number of connections of each sample. O_z is semi-dense if C_{Oz} is greater than or equal to the global density thresholds. Finally, O_z is outlier if C_{Oz} is less than the global density thresholds. These conditions are represented by the inequalities (4), (5), (6), where $median(C_{O1}, ..., C_{Ok})$ and $average(C_{O1}, ..., C_{Ok})$ correspond respectively to the median and mean of the set of connections of C.

$$\begin{aligned} O_z dense \Leftrightarrow C_{Oz} \geq median(C_{O1}, ..., C_{Ok}) \\ \wedge C_{Oz} \geq average(C_{O1}, ..., C_{Ok}) \end{aligned} \tag{4}$$

$$\begin{aligned} O_z semi-dense \Leftrightarrow C_{Oz} \geq median(C_{O1}, ..., C_{Ok}) \\ \vee C_{Oz} \geq average(C_{O1}, ..., C_{Ok}) \end{aligned} \tag{5}$$

$$\begin{aligned} O_z outlier \Leftrightarrow C_{Oz} < median(C_{O1}, ..., C_{Ok}) \\ \wedge C_{Oz} < average(C_{O1}, ..., C_{Ok}) \end{aligned} \tag{6}$$

Algorithm 1 implements the creation of clusters with the global approach of DyClee-C. From line 1 to 3, the density thresholds $median(C_{O1}, ..., C_{Ok})$ and $average(C_{O1}, ..., C_{Ok})$ are calculated from the set of connections of all the samples of C. When a dense sample is detected, his neighbors are searched for and analyzed. If these are dense neighbors of neighbors are sought. If the sample is semi-dense, the sample is added to the cluster but his neighbors are not sought. This approach allows for a cluster with a dense center and semi-dense cluster edges. A cluster is created when no new neighbor is detected. The operation continues with a new sample. Algorithm stops when all the samples have been analyzed (lines 4 to 24).

Algorithm 1 DyClee-C global approach

```
Require: Set of samples $O$, set of connexions of samples $C$
 1: $O_z \rightarrow Dense$ such that $C_{Oz} \geq median(C_{O1}, ..., C_{Ok}) \wedge C_{Oz} \geq average(C_{O1}, ..., C_{Ok})$
 2: $O_z \rightarrow Semi-dense$ such that $C_{Oz} \geq median(C_{O1}, ..., C_{Ok}) \vee C_{Oz} \geq average(C_{O1}, ..., C_{Ok})$
 3: $Clusters$ = []
 4: while $O$ not empty do
 5:   $toAnalyze$ = []
 6:   $cluster$ = []
 7:   $toAnalyze \leftarrow O.pop()$
 8:   while $toAnalyze$ not empty do
 9:     for $i \in toAnalyze$ do
10:       if $i \rightarrow Dense$ then
11:         $Neighbors \leftarrow LSH.query(i)$
12:         for $neighbor \in Neighbors$ do
13:           if ($neighbor$ is $Dense \wedge neighbor \notin cluster$) then
14:             $cluster.add(neighbor)$
15:             $toAnalyze.add(neighbor)$
16:           else if ($neighbor$ is $Semi-dense \wedge neighbor \notin cluster$) then
17:             $cluster.add(neighbor)$
18:           end if
19:         end for
20:       end if
21:     end for
22:   end while
23:   $Clusters.add(cluster)$
24:   $O \leftarrow$ samples that are not in $Clusters$
25: end while
```

To illustrate the global approach, two groups found by the LSH, are shown in Figure 4. The first categorical group is made up of eight samples G_1 =

$\{1, 2, 3, 4, 5, 6, 7, 8\}$ and the second is composed of six samples $G_2 = \{9, 10, 11, 12, 13, 14\}$. The set $C = \{5, 3, 1, 3, 2, 1, 2, 1, 2, 5, 2, 3, 3, 1\}$ corresponds to the connections of all samples. The overall density thresholds, i.e. the median and the mean of the connections present in C, are respectively equal to $median(C_{O1}, ..., C_{Ok}) = 2$ and $average(C_{O1}, ..., C_{Ok}) = 2.43$. A cluster is composed of dense samples in the center and semi-dense samples at the cluster boundaries.

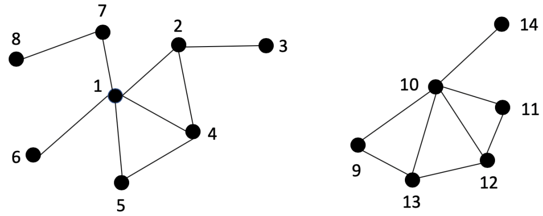

Figure 4: Example of groups of categorical samples detected by the LSH algorithm

For the group C_{G1}, the dense samples are $Dense\_C_{G1} = \{1, 2, 4\}$ and the semi-dense samples are $Semi - dense\_C_{G1} = \{5, 7\}$. Samples 1, 2, and 4 are dense because their number of C_{O1}, C_{O2}, and C_{O4} connections are greater than or equal to the overall density thresholds. Samples 5 and 7 are semi-dense because their number of connections C_{O5} and C_{O7} is greater than or equal to one of the global density thresholds. Samples 3, 6, and 8 are outliers because their number of C_{O3}, C_{O6}, and C_{O8} connections are less than the overall density thresholds. The same reasoning is applied for the second group. The resulting clusters are $Cl_1 = \{1, 2, 4, 5, 7\}$ and $Cl_2 = \{9, 10, 11, 12, 13\}$. The clusters found by applying the global approach are shown in Figure 5.

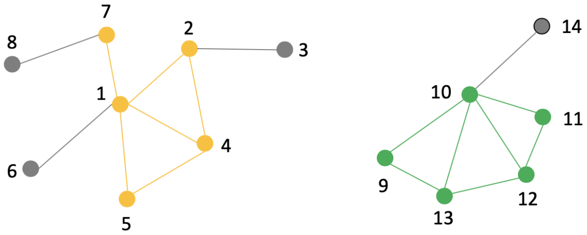

Figure 5: Clusters found by applying the global approach

In the **local approach** of DyClee, the density of a D_z micro-cluster is no longer compared to the median and mean density of all μCs micro-clusters, but to the median density and average of the micro-clusters of the group in which μC_z belongs. Let $D_G = \{D_{G1}, ..., D_{Gz}, ..., D_{Gl}\}$, all the densities of the groups found by the KD- Tree, D_{Gz}, the densities of the micro-clusters present in the group G_z and l the number of groups of micro-clusters. μC_z is dense if D_z is greater than or equal to the local density thresholds, i.e the median and mean density of the micro-clusters of the group D_{Gz} in which μC_z is included. μC_z is semi-dense if D_z is greater than or equal to one of the local density thresholds. Finally, μC_z is outlier if D_z is strictly less than the local density thresholds. These conditions are represented by the inequalities (7), (8), (9), where $median(D_{Gz})$ and $average(D_{Gz})$ correspond respectively to the median and mean density of micro-clusters belonging to the group G_z.

$$\mu D_z dense \Leftrightarrow D_z \geq median(D_{Gz}) \wedge D_z \geq average(D_{Gz}) \tag{7}$$

$$\mu D_z semi - dense \Leftrightarrow D_z \geq median(D_{Gz}) \vee D_z \geq average(D_{Gz}) \tag{8}$$

$$\mu D_z outlier \Leftrightarrow D_z < median(D_{Gz}) \wedge D_z < average(D_{Gz}) \tag{9}$$

In DyClee-C, the principle of creating clusters by local approach is unchanged but instead of comparing a sample's connections to the median and the average of the connections of all samples O, C_{Oz} is compared to the connection thresholds calculated from the median and the average of the connections of samples in the C_{Gz} group. The conditions for a sample to be dense, semi-dense and outlier are given by inequalities (10), (11) and (12). The pseudo code associated with the local approach is given by Algorithm 2.

$$O_z dense \Leftrightarrow C_{Oz} \geq median(C_{Gz}) \wedge C_{Oz} \geq average(C_{Gz}) \tag{10}$$

$$O_z semi - dense \Leftrightarrow C_{Oz} \geq median(C_{Gz}) \vee C_{Oz} \geq average(C_{Gz}) \tag{11}$$

$$O_z outlier \Leftrightarrow C_{Oz} < median(C_{Gz}) \wedge C_{Oz} < average(C_{Gz}) \tag{12}$$

To illustrate this approach, the example of Figure 6 is used. The density thresholds change in relation to the overall approach. Each group has its own density thresholds. Group 1 composed of $G_1 = \{1, 2, 3, 4, 5, 5, 6, 7, 8\}$ has for $median(D_{G1}) = 2$ and $average(D_{G1}) = 2{,}25$ and group 2 composed of $G_2 = \{9, 10, 11, 12, 12, 13, 14\}$ has for $median(D_{G2}) = 2.5$ and $average(D_{G2}) = 2.75$. The Cl_1 cluster remains unchanged from the overall approach. Concerning the Cl_2 cluster, samples 9 and 11 are added to sample 14 and are also considered outliers because C_{O9} and C_{O11} are less than $median(D_{G2})$ and $average(D_{G2})$. The Cl_2 cluster is therefore composed of $Cl_2 = \{10, 12, 13\}$. Figure 6 illustrates the clusters found by applying the local approach.

The methods for generating clusters in DyClee-C have been detailed in the previous section. Several parameters allow the user to improve clustering results by adding some knowledge to the data. These parameters are detailed below.

Algorithm 2 DyClee-C local approach

```
Require: Set of samples $O$, set of connexions of samples $C$, Set of groups of samples $C_G$
$O_z \rightarrow Dense$ such that $C_{Oz} \geq median(C_{Gz}) \wedge C_{Oz} \geq average(C_{Gz})$
$O_z \rightarrow Semi-dense$ such that $C_{Oz} \geq median(C_{Gz}) \vee C_{Oz} \geq average(C_{Gz})$
$Clusters$ = []
while $O$ $not$ $empty$ do
    $toAnalyze$ = []
    $cluster$ = []
    $toAnalyze \leftarrow O.pop()$
    while $toAnalyze$ $not$ $empty$ do
        for $i \in toAnalyze$ do
            if $i \rightarrow Dense$ then
                $Neighbors \leftarrow LSH.query(i)$
                for $neighbor \in Neighbors$ do
                    if $(neighbor$ $is$ $Dense \wedge neighbor \notin cluster)$ then
                        $cluster.add(neighbor)$
                        $toAnalyze.add(neighbor)$
                    else if $(neighbor$ $is$ $Semi-dense \wedge neighbor \notin cluster)$ then
                        $cluster.add(neighbor)$
                    end if
                end for
            end if
        end for
    end while
    $Clusters.add(cluster)$
    $O \leftarrow samples$ $that$ $are$ $not$ $in$ $Clusters$
end while
```

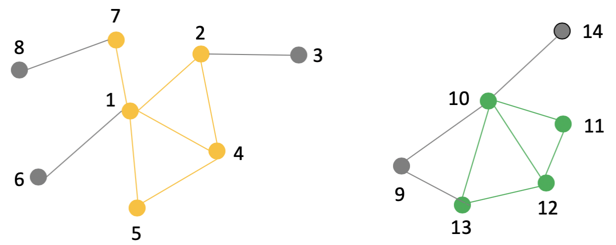

Figure 6: Clusters found by applying the local approach

3.3 Settings of Dyclee-C

The final clusters are found using one of the two approaches described in the previous section. These results can be refined with optional parameters. In this section, three parameters are described. The first is called $Unclassed\_accepted$, the second is called $minimum\_samples$ and the last parameter presented is called $n\_clusters$. These parameters are tuned in function of the problem encountered and evaluated with cluster validity methods [18] like those they are presented in section 4.

In DyClee, clusters are composed of dense (center) and semi-dense (edge of the cluster) micro-clusters. Outliers are considered unrepresentative and their samples are rejected (considered noise). Depending on the context, it may be interesting not to consider outliers but, on the contrary, to assign all the data to a cluster. When the parameter $Unclassed\_accepted$ is activated, all the samples must be assigned to a cluster, i.e. there is no outlier rejection.

The second parameter is called $minimum\_samples$ and allows you to set the minimum number of samples that a cluster must contain to be considered as a final cluster. Indeed, depending on the application, small clusters may not be representative. In the case where a nominal situation has to be analysed, small clusters can represent abnormal situations. These clusters are no longer considered as final clusters and the samples assigned to them are marked as noise. The equation (13) allows you to set the size that clusters must have to be considered as final clusters.

$$|Cl_z| > \frac{\sum_{i=0}^{l} |Cl_i|}{|Cl|} \tag{13}$$

with $|Cl_z|$ the size of the cluster z to be evaluated and $|Cl|$ is the number of cluster groups.

The use of the parameter is illustrated in Figure 7 with three clusters Cl = $\{Cl_1, Cl_2, Cl_3\}$. Cluster 1 is given by $Cl_1 = \{1, 2, 3, 4, 5, 6, 7, 8\}$, cluster 2 by $Cl_2 = \{9, 10, 11, 12, 13, 14\}$ and the third by $Cl_3 = \{15, 16, 17\}$. To be considered as final clusters, all Cl clusters must have a size greater than the threshold defined in the equation (14):

$$Threshold = \frac{|Cl_1| + |Cl_2| + |Cl_3|}{|Cl|} = \frac{8+6+3}{3} = 5,67 \tag{14}$$

The Cl_1 and Cl_2 clusters are larger than 5.67 (respectively 8 and 6) and are therefore considered final clusters. On the other hand, since the Cl_3 cluster has a size equal to 3, the samples that compose it (15, 16 and 17) are considered as noise (in the Figure 7 in grey).

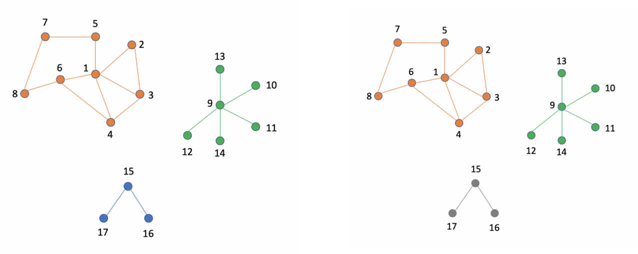

Figure 7: Left: parameter $minimum\_samples$ disabled. Right: parameter $minimum\_samples$ enabled

The last parameter is called $n\_clusters$ and allows you to consider the most important n clusters as final clusters. Samples belonging to the remaining clusters are assigned to the final n clusters. Let be A = $\{A_1, ..., A_i, ..., A_m\}$, all the categorical attributes of the set of samples O, with m the number of attributes. c = $\{c_1, ..., c_z, ..., c_l\}$ is the set of cluster centers Cl with c_z the cluster center Cl_z. A c_z center is defined by c_z = $\{c_{z1}, ..., c_{zi}, ..., c_{zm}\}$ with c_{zi}, the i-th component of the z center. The terms of an attribute A_i are noted $Mod(A_i)$ = $\{m_{i1}, ..., m_{ij}, ..., m_{ip}\}$ with m_{ij}, the j-th term of the attribute A_i and p, the number of terms of the attribute A_i. The frequency of a modality m_{ij} is noted $fr(m_{ij})$. The i component of the c_z center is defined as the most frequent modality of the A_i attribute (see equation (15).

$$c_{zi} = \max(fr(m_{i1}, ..., fr(m_{ij}), ..., fr(m_{ip})). \tag{15}$$

When the $n\_clusters$ parameter is active, the distance between samples not in the final n clusters is calculated and samples are assigned to the nearest center. Figure 8 takes the example of Figure 7. In this example, the two most important clusters are considered as final clusters (Cl_1 and Cl_2). Samples 15, 16 and 17 of Cl_3 are reassigned to the nearest clusters. Samples 15 and 17 are therefore assigned to the Cl_1 cluster and sample 16 to the Cl_2 cluster.

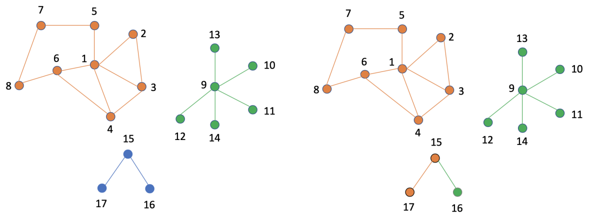

Figure 8: Left: parameter $n\_clusters$ disabled. Right: parameter $n\_clusters$ enabled with n = 2

The section 4 presents the different tests performed on the DyClee-C.

4 Evaluation

DyClee-C was tested on well-known UCI Machine Learning data sets in the clustering domain like *Zoo*, *Congressional Voting Records*, *Soybean* and *Breast Cancer*. The two latter data sets have been selected to be reported in this paper because they correspond to a diagnosis problem. As they have a large number of dimensions, it allows to test how DyClee-C handles data sets with many categorical attributes. DyClee-C was compared to the K-modes clustering algorithm [5]. Note that an initialization method for the centers of clusters that is not present in the Huang paper is also tested. Indeed, this new method was introduced by [19] 11 years after the original paper. Training phases have been realized on a part of the data sets to find correct combinations of parameters. To evaluate clusters, three validity measures are used: the *purity*, *recall* and the *precision*. Purity is the ratio between the sum of the number of elements correctly assigned in each class i (noted TP_i) and the number of samples in the data set. This measure is described in the equation (16) where TPi is the real positive rate of the i^{th} class, k is the number of classes and N is the number of samples in the data set.

$$Purity = \frac{\sum_{i=0}^{k} TP_i}{N} \tag{16}$$

The recall corresponds to the ratio between the number of elements correctly assigned to the i^{th} class and the number of elements belonging to the i^{th} class (noted TP_i + FN_i). This measure is described in the equation (17) with R_i, the recall of the i^{th} class, TP_i, the real positive rate in the i^{th} class and FN_i class, the false negative rate in the i^{th} class. A false negative is a result where the model incorrectly predicts the negative class.

$$R_i = \frac{TP_i}{TP_i + FN_i} \tag{17}$$

The last measure is the precision which corresponds to the ratio between the number of elements correctly assigned to the i^{th} class and the number of elements assigned to the i^{th} class (noted TP_i + FP_i). This measure is described in the equation (18), with P_i, the accuracy of the i^{th}, TP_i class, the true positive rate for the i^{th} and FP_i class, the false positive rate for the i^{th} class. A false positive is a result where the model incorrectly predicts the positive class.

$$P_i = \frac{TP_i}{TP_i + FP_i} \tag{18}$$

4.1 Soybean data set

The first data set is the "Soybean" [20]. This data set consists of 47 soybean plants with 35 categorical attributes and 4 classes. Attributes correspond to the characteristics of the plant (size of the seed, leaves,...). The classes correspond to diseases specific to soybean plants. Two tests with two different settings were performed on this data set. The first one was done with the parameter $Unclassed\_accepted$ enabled. The parameter of K-Modes corresponding to the number of clusters is $k\_clusters = 4$. Original classes of the "Soybean" data set, results of DyClee-C and K-Modes algorithms with both initializations [5], [19]) are shown in Table 1.

	C_1	C_2	C_3	C_4
Soybean	17	10	10	10
DyClee-C	26	10	10	1
K-Modes (Huang)	16	14	10	7
K-Modes (Cao)	**17**	**10**	**10**	**10**

Table 1: Clusters found by DyClee-C and K-Modes for the Soybean data set

Results in Table 1 show that classes C_2 and C_3 were perfectly detected by both algorithms tested. However, the C_1 and C_4 clusters were poorly formed by DyClee-C while K-Modes found the right classes. The reason behind the DyClee-C misclassification is that DyClee-C do not compute the distance between every samples. Indeed, the Locality Sensitive Hashing consider as nearest neighbors samples that share the same bucket at least once (section 3.1). As K-Modes method measures the dissimilarity between the whole data set, the Huang's algorithm is more precise. Moreover, K-Modes method needs the number of clusters as a parameter. As an unsupervised clustering algorithm, this kind of information is normally not known in advance. Purity, recall and precision measures for DyClee-C and K-Modes algorithms are shown in Table 2.

	Purity	Recall	Precision
DyClee-C	70%	77%	85%
K-Modes (Huang)	78%	55%	52%
K-Modes (Cao)	**100%**	**100%**	**100%**

Table 2: Purity, recall and precision scores of DyClee-C and K-Modes algorithm for the Soybean data set

As K-Modes method found correct classes, purity, recall and precision measures corresponding are equal to 100%. The second test highlights outliers and their influence on DyClee-C's results. Parameters enabled for this test are $Unclassed\_accepted$ and $minimum\_samples$. The latter makes it possible to eliminate samples belonging to clusters with a size smaller than the average cluster size.

	C_1	C_2	C_3	C_4
Soybean	10	9	9	6
Result	10	9	9	6

Table 3: Clusters found by DyClee-C for the Soybean data set with outliers removed

Table 3 represents the classes of the "Soybean" data set and the clusters found by DyClee-C when the outliers detected by the parameter $minimum\_samples$ are not taken into account. The results show that in this configuration, all classes were found by DyClee-C. The three validity measures associated to this result are equal to 100%.

4.2 Breast Cancer data set

The second data set is the "Breast Cancer" [21]. This data set consists of 286 instances and 9 categorical attributes and 2 classes. An instance corresponds to a patient and each attribute is an information about the patient and his pathology (age, which breast has the tumor, the degree of the malignancy,...). Some attributes have missing values and are represented by a "?" in the data set. These special values are considered as a modality for the test. Classes are instances which have a no recurrence events and recurrence events. Parameters enabled of DyClee-C for this test are $Unclassed\_accepted$ and $n\_clusters = 2$. The parameter of K-Modes corresponding to the number of clusters is $k\_clusters = 2$. Original classes of the "Breast Cancer" data set, results of DyClee-C and K-Modes (with both initializations) are shown in Table 4.

	C_1	C_2
Breast Cancer	201	85
DyClee-C	**207**	**79**
K-Modes (Huang)	149	137
K-Modes (Cao)	183	103

Table 4: Clusters found by DyClee-C and K-Modes for the Breast Cancer data set

DyClee-C have detected better groups than the K-Modes algorithm. Few samples have been misclassified by DyClee-C while they are more samples in the wrong class with the K-Modes algorithm. Purity, recall and precision measures for DyClee-C and K-Modes algorithms are shown in Table 5.

	Purity	Recall	Precision
DyClee-C	**72%**	**63%**	**63%**
K-Modes (Huang)	70%	45%	45%
K-Modes (Cao)	70%	53%	52%

Table 5: Purity, recall and precision scores of DyClee-C and K-Modes algorithm for the Breast Cancer data set

While purity score is slightly higher for DyClee-C than K-Modes, the method presented in this paper is clearly better for recall and precision score than the Huang's algorithm.

5 Conclusions and perspectives

In this article, an extension of DyClee, a dynamic clustering algorithm for digital data, is presented. This extension, called DyClee-C, allows you to apply the basic concepts of DyClee to categorical data. The KD-Tree has been replaced by the LSH algorithm, which makes it possible to form groups of categorical samples with large dimensions. The two approaches, global and local density, to generate clusters have been modified to capture categorical data. The concept of micro-cluster density is replaced by the number of neighbours of a sample, i.e. connections. Thus, a sample with a certain number of connections is more dense than a sample with few connections. Three parameters have been adapted in order to refine the clusters obtained. The first one allows you to have no noise, the second one only keeps clusters with a size larger than the average cluster size and the last one allows you to consider the most important n clusters as final clusters.

For the tests, DyClee-C has been compared to the K-Modes algorithm on two well-known data sets. The *Breast Cancer* data set shows that DyClee-C is able to detect classes even if the samples have missing modalities. The *Soybean* data sets illustrates the capacity of DyClee-C to detect outliers. The obtained results are promising and show that clustering can be used for diagnosis purposes even for data bases with categorical features. However, as classes were known in advance, results are biaised. Indeed, an unsupervised classification algorithm is applied to a dataset which labels are not known. Subsequently, further tests and comparisons with other algorithms are to be carried out.

In the future, we plan to develop a mixed version of DyClee to manage both numerical and categorical data.

References

[1] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[2] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[3] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[4] Gabor J Szekely and Maria L Rizzo. Hierarchical clustering via joint between-within distances: Extending ward's minimum variance method. *Journal of classification*, 22(2):151–183, 2005.

[5] Zhexue Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values.

Data mining and knowledge discovery, 2(3):283–304, 1998.

[6] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Rock: A robust clustering algorithm for categorical attributes. *Information systems*, 25(5):345–366, 2000.

[7] Zengyou He, Xiaofei Xu, and Shengchun Deng. Squeezer: an efficient algorithm for clustering categorical data. *Journal of Computer Science and Technology*, 17(5):611–624, 2002.

[8] Nathalie Barbosa Roa, Louise Travé-Massuyès, and Victor H. Grisales-Palacio. Dyclee: Dynamic clustering for tracking evolving environments. *Pattern Recognition*, 2019, https://doi.org/10.1016/j.patcog.2019.05.024.

[9] Nathalie Andrea Barbosa Roa. *A data-based approach for dynamic classification of functional scenarios oriented to industrial process plants*. PhD thesis, Université de Toulouse, Université Toulouse III-Paul Sabatier, 2016.

[10] Nathalie Barbosa, Louise Travé Massuyes, and Victor Hugo Grisales. A data-based dynamic classification technique: A two-stage density approach. In *SAFEPROCESS 2015, Proceedings of the 9th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes*, pages 1224–1231. IFAC, 2015.

[11] Songrit Maneewongvatana and David M Mount. ItâĂŹs okay to be skinny, if your friends are fat. In *Center for Geometric Computing 4th Annual Workshop on Computational Geometry*, volume 2, pages 1–8, 1999.

[12] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613. ACM, 1998.

[13] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge university press, 2014.

[14] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. Similarity search in high dimensions via hashing. In *Vldb*, volume 99, pages 518–529, 1999.

[15] Jerome H Friedman. On bias, variance, 0/1—loss, and the curse-of-dimensionality. *Data mining and knowledge discovery*, 1(1):55–77, 1997.

[16] John Rust. Using randomization to break the curse of dimensionality. *Econometrica: Journal of the Econometric Society*, pages 487–516, 1997.

[17] Michel Verleysen and Damien François. The curse of dimensionality in data mining and time series prediction. In *International Work-Conference on Artificial Neural Networks*, pages 758–770. Springer, 2005.

[18] Yanchi Liu, Zhongmou Li, Hui Xiong, Xuedong Gao, and Junjie Wu. Understanding of internal clustering validation measures. In *2010 IEEE International Conference on Data Mining*, pages 911–916. IEEE, 2010.

[19] Fuyuan Cao, Jiye Liang, and Liang Bai. A new initialization method for categorical data clustering. *Expert Systems with Applications*, 36(7):10223–10228, 2009.

[20] Ryszard S Michalski. UCI machine learning repository, 1987.

[21] Min Tan and Jeff Schlimmer. UCI machine learning repository, 1988.