Computer-aided Diagnosis via Hierarchical Density Based Clustering
Tom Obry, Louise Travé-Massuyès, Audine Subias

To cite this version:
Tom Obry, Louise Travé-Massuyès, Audine Subias. Computer-aided Diagnosis via Hierarchical Density Based Clustering. 29th International Workshop on Principles of Diagnosis (DX 2018), Aug 2018, Varsovie, Poland. 8p. hal-01847563

HAL Id: hal-01847563
https://hal.laas.fr/hal-01847563
Submitted on 23 Jul 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Computer-aided Diagnosis via Hierarchical Density Based Clustering

Tom Obry\textsuperscript{1,2} and Louise Trévé-Massuyès \textsuperscript{1} and Audine Subias \textsuperscript{1} \\
\textsuperscript{1}LAAS-CNRS, Université de Toulouse, CNRS, INSA, Toulouse, France \\
\textsuperscript{2}ACTIA, 5 Rue Jorge Semprun, 31432 Toulouse

Abstract

When applying non-supervised clustering, the concepts discovered by the clustering algorithm hardly match business concepts. Hierarchical clustering then proves to be a useful tool to exhibit sets of clusters according to a hierarchy. Data can be analyzed in layers and the user has a full spectrum of clusterings to which he can give meaning. This paper presents a new hierarchical density-based algorithm that advantageously works from compacted data. The algorithm is applied to the monitoring of a process benchmark, illustrating its value in identifying different types of situations, from normal to highly critical.

1 Introduction

In data-based diagnosis applications, it is often the case that huge amounts of data are available but the data is not labelled with the corresponding operating mode, normal or faulty. Clustering algorithms, known as non-supervised classification methods, can then be used to form clusters that supposedly gather data corresponding to the same operating mode.

Clustering is a Machine Learning technique used to group data points according to some similarity criterion. Given a set of data points, a clustering algorithm is used to classify each data point into a specific group. Data points that are in the same group have similar features, while data points in different groups have highly dissimilar features. Among well-known clustering algorithms, we can mention K-Means [1], PAM [2], K-Modes [3], DBSCAN [4].

Numerous validity indexes have been proposed to evaluate clusterings [5]. These are generally based on two fundamental concepts:

- compactness, the members of each cluster should be as close to each other as possible. A common measure of compactness is the variance, which should be minimized.
- separation, the clusters themselves should be widely spaced.

Nevertheless, one must admit that the concepts discovered by even the most scored clusterings hardly match business concepts [6] [7]. One of the reasons is that data bases are often incomplete in the sense that they do not include the data about all the influential attributes. In particular, business concepts are highly sensitive to environmental parameters that fall outside the scope of the considered business domain and that are not recorded, for instance stock exchange. In addition, the clusters corresponding to business concepts may be quite “close” in the data space and the only way to capture them would be to guess the right number of clusters to initialize correctly the clustering algorithm. This is obviously quite hard. Hierarchical clustering then proves to be a useful tool because it exhibits sets of clusters according to a hierarchy and it modulates the number of clusters. Data can then be analyzed in layers, with a different number of clusters at each level, and the user has a full spectrum of clusterings to which he can give meaning.

Hierarchical clustering identifies the clusters present in a dataset according to a hierarchy [8][9][10]. There are two strategies to form clusters, the agglomerative (“bottom up”) strategy where each observation starts in its own cluster and pairs of clusters are merged as one moves up the hierarchy. The divisive method (“top down”) where all observations start in one cluster and splits are performed recursively as one moves down the hierarchy. The results of hierarchical clustering are usually presented in a dendrogram. A dendrogram is a tree diagram frequently used to illustrate the arrangement of the clusters. In order to decide which clusters should be combined or where a cluster should be split, a measure of dissimilarity between sets of observations is required. In most methods of hierarchical clustering, splits or merges of clusters are achieved by use of an appropriate metric like euclidean, manhattan or maximum distance.

Few algorithms propose a density-based hierarchical clustering approach like n-unchaining single linkage [11] or HDBSCAN [12]. In this paper, we present a new hierarchical clustering algorithm, named HDycle, based on density that advantageously works from compacted data in the form hypercubes. This contribution is an extension of the clustering algorithm DyClee [13], [14], [15]. The purpose of this work is to generate a flat partition of clusters with a hypercube’s density level higher or equal to a threshold and to be able to visualize all existant clusters in the dataset with a dendogram by varying the density of the hypercubes present in a group. The value of the algorithm in a diagnosis context is illustrated with the monitoring of a Continuous Stirred Tank Heater benchmark, for which it allows the user to identify different types of situations, from normal to highly critical.

This paper is organized as follows. In section 2 the Dy-Clee algorithm is presented. In section 3 the concepts and
principles underlying DyClee, like the definition of micro clusters $\mu C$, dynamic clusters and the KD-Tree structure, are explained. In the section 4, the hierarchical clustering based-density algorithm is presented. Tests and results are detailed in section 5. The conclusion and perspective for future work end this paper in section 6.\textsuperscript{1}

2 DyClee: a dynamic clustering algorithm

DyClee is a dynamic clustering algorithm which is able to deal with large amounts of data arriving at fast rates by adopting a two stages strategy similar to [16], [17], [18]. The first stage is a fast scale distance-based algorithm that collects, pre-processes and compresses data samples to form so-called micro-clusters ($\mu$-clusters). It operates at the rate of the data stream and creates $\mu$-clusters putting together data samples that are close, in the sense of a given distance, to each other. $\mu$-clusters are stored in the form of summarized representations including statistical and temporal information.

The second stage is a slower scale density-based algorithm that groups the $\mu$-clusters into actual clusters that can be interpreted semantically as classes. It takes place once each $t_{\text{slow}}$ seconds and analyses the distribution of $\mu$-clusters. The density of a $\mu$-cluster is considered as low, medium or high and is used to create the final clusters by a density based approach, i.e., dense $\mu$-clusters that are close enough (connected) are said to belong to the same cluster. Similarly to [19], a cluster is defined as the group of connected $\mu$-clusters where every inside $\mu$-cluster presents high density and every outside $\mu$-cluster exhibits either medium or low density. The above dense $\mu$-cluster structure allows the algorithm to create clusters of non convex shapes even in high dimensional spaces and it has proved outliers rejection capabilities in evolving environments [18]. In addition, $\mu$-clusters of similar densities can form clusters of any shape and any size.

In DyClee, both stages work on-line, but operate at different time scales. This multi-density feature allows the detection of novelty behavior in its early stages when only a few objects giving evidence of this evolution are present. Figure 1 gives the global description of DyClee.

3 Main principles of DyClee

All the principles explained in this section are from the core algorithm.

3.1 Notion of micro-clusters $\mu C$

Considering a $d$-dimensional object $X = [x^1, \ldots, x^d]$ marked with a timestamp $t_X$ and qualified by $d$ features, a $\mu$-cluster gathers a group of data samples close in all dimensions and whose information is summarized in a characteristic feature vector (CF). For a $\mu$-cluster $\mu C_k$, CF has the following form:

$$ CF_k = (n_k, L_{S_k}, S_{S_k}, l_{t_k}, t_{s_k}, D_k, Class_k). \quad (1) $$

where $n_k \in \mathbb{N}$ is the number of objects in the $\mu$-cluster $k$, $L_{S_k} \in \mathbb{R}^d$ is the vector containing the linear sum of each feature over the $n_k$ objects, $S_{S_k} \in \mathbb{R}^d$ is the square sum of features over the $n_k$ objects, $l_{t_k} \in \mathbb{R}$ is the time when the last object was assigned to that $\mu$-cluster, $t_{s_k} \in \mathbb{R}$ is the time when the $\mu$-cluster was created, $D_k$ is the $\mu$-cluster density and $\text{Class}_k$ is the $\mu$-cluster label if known. Using $L_{S_k}, S_{S_k}$ and $n_k$ the variance of the group of objects assigned to the $\mu$-cluster $k$ can be calculated.

The $\mu$-cluster is shaped as a $d$-dimensional box since the L1-norm is used as distance measure. The distance between an object $X = [x^1, \ldots, x^d]$ and a $\mu$-cluster $\mu C_k$, named as $\text{dis}(X, \mu C_k)$, is calculated as the sum of the distances between the $\mu C_k$ vector center $c_k = [c^1_k, \ldots, c^d_k]^T$ and the object value for each feature as shown in equation (2):

$$ \text{dis}(X, \mu C_k) = L_1(X, c_k) = \sum_{i=1}^d |x^i - c^i_k|. \quad (2) $$

The data is normalized according to the data context, i.e. the feature range $[\text{min}^i, \text{max}^i]$ of each feature $i$, $i = 1, \ldots, d$. If no context is available in advance, it may be established online. The size of the hyperboxes $S_i$ along each dimension $i$ is set as a fraction of the corresponding feature range. The hyperbox size per feature is hence found according to (3), where $\phi$ is a user constant parameter in the interval $(0, 1)$, establishing the fraction:

$$ S_i = \phi(|\text{max}^i - \text{min}^i|), \quad \forall i = 1, \ldots, d. \quad (3) $$

Whenever an object $X$ arrives, the algorithm searches for the closest $\mu$-cluster. Once found, a maximal distance criterion is evaluated to decide whether or not the object fits inside the $\mu$-cluster hyper-box. If the fitting is sufficient the $\mu$-cluster $\mu C_k$ is created, $D_k$ is the $\mu$-cluster density and $\text{Class}_k$ is the $\mu$-cluster label if known. Using $L_{S_k}, S_{S_k}$ and $n_k$ the variance of the group of objects assigned to the $\mu$-cluster $k$ can be calculated.

The density of a $\mu$-cluster $\mu C_k$ is calculated using the current number of objects $n_k$ and the current hyper-volume of the bounding box $V_k = \prod_{i=1}^d S_i$, as shown in (4):

$$ D_k = \frac{n_k}{V_k}. \quad (4) $$

Let $\mu C_{k_1}$ and $\mu C_{k_2}$ be two $\mu$-clusters, then $\mu C_{k_1}$ and $\mu C_{k_2}$ are said to be directly connected if their hyper-boxes overlap in all but $\varphi$ dimensions, where $\varphi$ is an integer. The parameter $\varphi$, fixed by the user, establishes the feature selectivity.

A $\mu$-cluster $\mu C_{k_i}$ is said to be connected to $\mu C_{k_j}$ if there exists a chain of $\mu$-clusters $\{\mu C_{k_1}, \mu C_{k_2}, \ldots, \mu C_{k_n}\}$ such that $\mu C_{k_i}$ is directly connected to $\mu C_{k_{i+1}}$ for $i = 1, 2, \ldots, n - 1$. A set of connected $\mu$-clusters is said to be a group.

\textsuperscript{1}This work is performed in the framework of a CIFRE Project supported by ACTIA.
3.2 Dynamic clusters

Dyclee is a dynamic clustering algorithm, which means that not only the parameters but the classifier structure changes according to input data in an automatic way. It achieves several cluster operations like creation, elimination, drift, merge, and split. For instance, a cluster is splitted into two or more clusters if, with the arrival of new data, high density regions can be distinguished inside the cluster. In that scenario, dense regions are separated by low density regions, making the cluster no longer homogeneous. Even more, the cluster center could be situated in a low density region, loosing its interpretability as prototype of the elements in the cluster. Splitting the cluster creates smaller homogeneous clusters, completely representative of the belonging samples. An illustrative example of this phenomenon is shown in Figure 2.

![Figure 2: Splitting of cluster A.](image)

3.3 Finding groups of \( \mu \)-clusters: the KD-Tree structure

To find groups of connected \( \mu \)-clusters, a KD-Tree [20] is used. A KD-Tree is a binary tree, each of whose nodes represents an axis-aligned hyperrectangle. Each node specifies an axis and splits the set of points based on whether their coordinate along that axis is greater than or less than a particular value. The tree is queried to return only neighbors who are at a maximum distance from a point. A \( \mu C_j \) is the neighbor of the \( \mu C_k \) if the condition in equation (5) is respected:

\[
L_{\infty} = \max_{i=1}^{d} |x_{ki}^i - c_j^i| < r. \tag{5}
\]

where \( d \) is the number of dimension, \( c_j^i \) the center of the \( \mu C_j \) at the dimension \( i \) and \( r \) the maximal distance from the \( \mu C_k \). In this context, \( r \) is set to \( \phi \).

4 A new hierarchical clustering density-based algorithm

In this section, a new hierarchical clustering density-based algorithm is presented. Inputs are the connections between \( \mu \)-clusters from the KD-Tree and the output is a flat partition of clusters where all \( \mu \)-clusters that are in clusters have a minimum density level guaranteed.

4.1 Representation of the \( \mu \)-clusters connections

The representation of the connections between all \( \mu \)-clusters can be visualized by a weighted Graph. A weighted Graph \( G = (V, E, W) \) is a triplet where \( V \) is a set of nodes. A node \( n_i \) corresponds to the \( \mu \)-cluster \( \mu C_i \). \( E \) is the set of edges with \( e_{ij} \), the edge between the node \( n_i \) and \( n_j \), which are unordered pairs of elements of \( G \). Finally, \( W \) is a set of weights on \( E \) with \( w_{ij} \) defined in the equation 6.

\[
w_{i,j} = \min(D_i, D_j). \tag{6}
\]

where the density of a \( \mu \)-cluster \( D_i \) is defined in the equation 4.

An edge \( e_{ij} \) between \( \mu \)-clusters \( \mu C_i \) and \( \mu C_j \) means those are directly connected in the sense defined in the section 3.1. If two \( \mu \)-clusters are not directly connected, there is no edge between them which leads to a Graph that is not full. The Graph of \( \mu \)-cluster’s connections is built according to the algorithm 1. Neighbors are searched for each \( \mu \)-cluster with respect to the equation 5 (lines 3 to 6). The function \( \text{Search_neighbors()} \) is detailed in the algorithm 2. For each \( \mu C_k \), the distance \( L_{\infty} \) defined in equation 5 is applied where \( r = \phi \), \( x_{ki}^i \) the value of \( \mu C_k \) at the \( i^{th} \) dimension and \( c_j^i \) the center of the \( \mu C_j \) at the dimension \( i \). The variable \( \text{Neighbors_of}_k \) contains all the neighbors of \( \mu C_k \). An edge \( e_{kj} \) is added to the Graph \( G \) for each neighbor \( \mu C_j \) of the \( \mu \)-cluster studied \( \mu C_k \) and the weight \( w_{k,j} \) is calculated with the equation defined above (lines 7 to 11).

Algorithm 1 Build the Graph of \( \mu \)-cluster’s connections

Require: KD-Tree
1: \( G = \text{Graph()} \)
2: Connection = ()
3: for \( k = 1 \) to Number of \( \mu \)-clusters do
4: \( N = [] \)
5: \( N = \text{Search_neighbors}(k) \)
6: Connection[k] = N
7: for \( j = 1 \) to Number of neighbors of \( k \) do
8: \( \text{Weight} = \min(D_k, D_j) \)
9: \( G.add\_edges(k, j, \text{Weight}) \)
10: end for
11: end for

Algorithm 2 Research of a \( \mu \)-cluster’s neighbors

Require: KD-Tree, \( \mu \)-cluster \( \mu C_k \)
1: \( \text{Neighbors\_of}_k = [] \)
2: for \( j = 1 \) to Number of \( \mu \)-clusters do
3: \( \text{Neighbors\_of}_k = \text{Search\_neighbors}(j) \)
4: \( \text{Neighbors\_of}_k.add() \)
5: end if
6: end for
7: return \( \text{Neighbors\_of}_k \)

Let us consider five \( \mu \)-clusters \( \mu C_1, \mu C_2, \mu C_3, \mu C_4 \) and \( \mu C_5 \) with their densities \( D_1 = 14, D_2 = 12, D_3 = 3, D_4 = 9, D_5 = 13 \). Figure 3 shows those five \( \mu \)-clusters.

![Figure 3: Example of five \( \mu \)-clusters.](image)
The search of neighbors begins with the densest $\mu$-cluster. In this example, the research starts with the $\mu C_1$ as shown in Figure 4. All $\mu$-clusters that satisfy the equation 5 are neighbors of $\mu C_1$.

![Figure 4: Search of neighbors for the $\mu C_1$.](image)

Once neighbors of $\mu C_1$ are found, the neighbors of the other $\mu$-clusters are searched. The result is shown in table 1:

<table>
<thead>
<tr>
<th>$\mu$-clusters</th>
<th>Neighbors</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu C_1$</td>
<td>$\mu C_2$</td>
</tr>
<tr>
<td>$\mu C_2$</td>
<td>$\mu C_1$, $\mu C_3$</td>
</tr>
<tr>
<td>$\mu C_3$</td>
<td>$\mu C_2$, $\mu C_4$</td>
</tr>
<tr>
<td>$\mu C_4$</td>
<td>$\mu C_3$, $\mu C_5$</td>
</tr>
<tr>
<td>$\mu C_5$</td>
<td>$\mu C_4$</td>
</tr>
</tbody>
</table>

Table 1: Neighbors for each $\mu$-clusters

The weights are calculated for every edge following the equation 6:

\[
\begin{align*}
  w_{12} &= \min(D_1, D_2) = \min(14, 12) = 12, \\
  w_{23} &= \min(D_2, D_3) = \min(12, 3) = 3, \\
  w_{34} &= \min(D_3, D_4) = \min(3, 9) = 3, \\
  w_{45} &= \min(D_4, D_5) = \min(9, 13) = 9.
\end{align*}
\] (7)

The Figure 5 shows the weighted Graph which represents the connections between all $\mu$-clusters in the dataset.

![Figure 5: Example 1: Graph of connections between $\mu$-clusters.](image)

4.2 Representation of the hierarchy of clusters

The objective of the algorithm proposed in this paper is to represent and visualize all the possible clusters at different levels of density in a dataset. In contrast to most known hierarchical clustering algorithm, links that relies level $l_n$ and $l_n + 1$ in our new algorithm’s dendogram are not based on the distance between objects but on their densities. Furthermore, our proposal is not based on the objects but on $\mu$-clusters that contains the objects to decrease the complexity of calculation. At the root of the tree, there is one cluster composed by all $\mu$-clusters. Each cut in the tree corresponds to a density threshold, i.e each cluster formed below this cut level is composed by $\mu$-clusters that have a density higher to the density threshold. At the bottom of the tree, there is one cluster for each $\mu$-cluster. So root’s density level is 0 and the last cut on the tree is $\max\{w \in \mathcal{W}\}$. Like [12], a variable $\epsilon$ is user parameter evolving in the interval $\epsilon \in [0, \max\{w \in \mathcal{W}\}]$. For each iteration of $\epsilon$, all the weights $w \in \mathcal{W}$ are checked. if $w_{ij} < \epsilon$, the edge $e_{ij}$ is removed. The vertical axis of the dendogram is $1/D$ to keep an ascending direction.

![Figure 6: Example 1: Graph of connections between $\mu$-clusters after removed all weight’s edges $w_{ij} < 3$.](image)

At the density threshold $D = 3$, two clusters composed by $\mu C_1$, $\mu C_2$ and $\mu C_4$, $\mu C_5$ are found and the $\mu$-cluster $\mu C_3$ is alone. One cluster composed by $\mu C_1$ and $\mu C_2$ is found and $\mu C_3$, $\mu C_4$. Then for $D=9$, one cluster composed by $\mu C_3$. Finally, above density $D = 12$, all $\mu$-clusters are alone.

4.3 Extracting a flat partition of clusters

The dendogram generated, a partition of clusters corresponding to a specific density level can be extracted from the Graph of $\mu$-cluster’s connections with a density threshold according to the algorithm 2. To find clusters, the edges that have a weight strictly lower than the density threshold are removed (lines 1 to 7). The remaining edges in the Graph have their weights higher or equal to the density threshold.

![Figure 7: Dendogram of the example 1.](image)
hence $\mu$-clusters forming the clusters are guaranteed to have their densities higher or equal to the threshold. Remaining groups are searched in the Graph. If the size of a group is equal to 1, i.e. the group have not connection with other $\mu$-clusters, it is considered as noise. Else, the group is recognized as a cluster (lines 10 to 16).

Figure 8: Example 2 : Graph of connections between eight $\mu$-clusters.

An other example is shown in Figure 8 to illustrate this part. Let us consider $\mu$-clusters $\mu C_i$, $i = 1, \ldots, 8$ with their respective densities $D_1 = 12$, $D_2 = 3$, $D_3 = 10$, $D_4 = 11$, $D_5 = 13$, $D_6 = 9$, $D_7 = 10$ and $D_8 = 4$.

Figure 9: Dendogram of the figure 8.

In this example, the density threshold is fixed to $\varepsilon = 8$. Every group of $\mu$-clusters that is below the density threshold is considered as a cluster. If a $\mu$-cluster does not have connection to the other $\mu$-clusters, then it is considered as noise. The Figure 10 illustrates how to visualize the clusters that have a density strictly above the threshold.

Figure 11: Clusters formed with a density guaranteed strictly above 8.

Algorithm 3

Extract a flat partition of clusters

Require: A density threshold $DT$, Weighted Graph $G$

1: for $\varepsilon = 0$ to $DT$ do
2: for all $w(i, j)$ in $G$ do
3: if $w(i, j) < \varepsilon$ then
4: $G$.removed(e(i,j))
5: end if
6: end for
7: end for
8: Groups = Find_Groups()
9: $k = 0$
10: for all $g$ in Groups do
11: if size($g$) == 1 then
12: $g$ = Noise
13: else
14: Cluster_$k = g$
15: end if
16: end for

5 Preliminary tests and results

HDyClee is tested on a benchmark similar to the well known Continuous Stirred Tank Heater (CSTH) of [21]. The CSTH is a stirred tank in which hot and cold water are mixed and further heated using steam. The final mix is drained using a long pipe [14]. Figure 12 shows the structure of the CSTH.

Process inputs are set-points for the cold water, hot water and steam valves. Process outputs are hot and cold water flow, tank level and temperature. Process inputs and outputs represent electronic signals in the range 4-20 mA. The test is done using three output variables: cold water flow $CW_{flow}$, tank level $Tank_{level}$, and temperature of the water in the tank $Tank_{temperature}$, in the operation mode OP1. In this mode, these variables are regulated at the values provided in table 2. The process undergoes several
faults and several repairs that are reported in table 3.

<table>
<thead>
<tr>
<th>Variable</th>
<th>OP1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CW_flow (mA)</td>
<td>11.89</td>
</tr>
<tr>
<td>Tank_level (mA)</td>
<td>12.00</td>
</tr>
<tr>
<td>Tank_temperature (mA)</td>
<td>10.50</td>
</tr>
</tbody>
</table>

Table 2: Nominal values for the test on CSTH.

The measurements of CW\_flow, Tank\_level, and Tank\_temperature are shown in Figure 13 and their recorded values across time constitute the data set for our hierarchical clustering experiment. Sudden changes in the value of regulated variables are indicative of the occurrence of some fault or of some fault being fixed. The dataset was generated by simulation.

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>l₁</td>
<td>Evolving leak starts. Hole diameter goes from 1 to 3, 5mm in 1500 seconds</td>
</tr>
<tr>
<td>l₁</td>
<td>Leak fixed</td>
</tr>
<tr>
<td>l₂</td>
<td>A second evolving leak starts. The second hole goes from 0 to 1mm in 1500 seconds</td>
</tr>
<tr>
<td>l₁l₂</td>
<td>Leaks fixed</td>
</tr>
<tr>
<td>s₁</td>
<td>Steam Valve stuck (closed)</td>
</tr>
<tr>
<td>s₁</td>
<td>Valve repaired</td>
</tr>
<tr>
<td>s₂</td>
<td>Hot water valve stuck at 10%</td>
</tr>
<tr>
<td>s₂</td>
<td>Valve repaired</td>
</tr>
<tr>
<td>l₃</td>
<td>Evolving leak starts. Hole goes from 1 to 2.6mm in 1000 seconds</td>
</tr>
<tr>
<td>l₁</td>
<td>Leak fixed</td>
</tr>
</tbody>
</table>

Table 3: Description of faults on the CSTH system for the operation mode OP1.

For this experiment, the radius of research for a \( \mu \)-cluster’s neighbors \( r \) is set to 0.06. The parameter \( \varphi \) that defines the number of dimensions that must overlap so that two \( \mu \)-clusters are considered directly connected is set to 0. That means two \( \mu \)-clusters \( \mu C_{k_a} \) and \( \mu C_{k_b} \) are directly connected if their hyperboxes overlap in all dimensions. The parameter \( \varepsilon \) is set to 0 for results shown in Figure 16 that correspond to the root of the dendogram. For the following graphs, the x-axis is the flow CW\_flow normalized and the y-axis is the tank temperature Tank\_temperature normalized. The tank level Tank\_level is not plotted. Figure 14 shows the graph of connections between \( \mu \)-clusters when \( \varepsilon = 0 \). Each red square represents a \( \mu \)-cluster. Micro-clusters that are not connected to the others are considered as noise. The \( \mu \)-cluster \( \mu C_{293} \) is connected to \( \mu C_{222} \) and \( \mu C_{232} \), meaning there are directly connected. \( \mu C_{222} \) is connected to \( \mu C_{34} \) and \( \mu C_{232} \) is connected with \( \mu C_{41} \) and so on. This chain of \( \mu \)-clusters forms a cluster. Figure 15 shows a generalized dendogram representing the hierarchy of behaviors found in the dataset.

Figure 13: Process measurements for multiple fault scenario.

Figure 14: Graph of \( \mu \)-clusters connections for the study case when \( \varepsilon = 0 \).
Figure 15: The dendogram with the nominal behavior and the occurrence of faults.

HDyClee detects the 6 main behaviors as shown the Figure 16. The biggest cluster (green cluster) represents the nominal behavior, the blue cluster represents fault $s_2$, the pink cluster (bottom of the Figure) represents the fault $s_1$, the grey cluster models the fault $l_1$, the brown cluster shows the event $l_1$, and the purple cluster represents $l_1$ and $l_2$ present simultaneously. Black points represent $\mu$-clusters that have no connection with other $\mu$-clusters. All the objects inside these $\mu$-clusters are considered as noise.

Figure 16: Clusters found by HDyClee algorithm: the nominal behavior (green cluster) and abnormal situations.

It is possible to visualize the most frequent behaviors of the system, in our case the normal behavior and fault $l_3$. This is represented in Figure 17. For this purpose, the density threshold is set to $\varepsilon = 6$ by using the dendogram. At this density, the clusters corresponding to other behaviors are considered as noise because their maximal densities are less than 6.

Figure 17: The nominal behavior (blue) and the most frequent fault (green).

To visualize only the nominal behavior, the dendogram must be cut at the density threshold $\varepsilon = 7$ because above this value, the other clusters have no $\mu$-clusters that are connected to each other. This is shown on Figure 19. Figure 18 illustrates the graph of $\mu$-cluster connections after deletion of the edges with a weight $w_{ij} < 7$. Some $\mu$-clusters which were part of the biggest cluster are now considered as noise. Indeed, edges that connected them to other $\mu$-clusters were less than the density threshold.

Figure 18: Graph of $\mu$-clusters connections for the study case after removing the weight’s edges $w_{ij} < 7$.

Figure 19: Visualization of the nominal behavior after having cut the dendogram at level $\varepsilon = 7$.

6 Conclusion and perspectives

The work presented in this paper proposes a new hierarchical density-based algorithm named HDyClee. The purpose of this algorithm is to extract a hierarchy of clusters that are guaranteed to have a level of density at each layer. Branches in the dendogram do not represent distance between objects but minimum density difference. This approach allows one to identify clusters with the poorest densities and then walk up the hierarchy for higher densities. The algorithm is detailed and tested on a well known monitoring benchmark. HDyClee is able to detect all the behaviors of the process and the user can explore more or less frequent behaviors by cutting the dendogram at different densities.

Next step is to develop experimentations in order to compare this new algorithm with other density-based algorithms. Then the comparative study will include hierarchical and distance-based clustering methods [22], [23]. Several perspectives have been identified for HDyClee, which follow from DyClee properties. In particular, DyClee has a forgetting function that allows to forget $\mu$-clusters which do not receive any data or are not significant (not dense $\mu$-clusters). This function will be included in HDyClee, which
will allow us to produce a dynamic dendogram and then to visualize the evolution of the different behaviors of a system.

References


