Overlapping Hierarchical Clustering (OHC)

Ian Jeantet, Zoltan Miklos, David Gross-Amblard

Overlapping Hierarchical Clustering (OHC)

Ian Jeantet, Zoltán Miklós, David Gross-Amblard

Univ Rennes, CNRS, IRISA, Rennes, France
first.last@irisa.fr

Abstract. Agglomerative clustering methods have been widely used by many research communities to cluster their data into hierarchical structures. These structures ease data exploration and are understandable even for non-specialists. But these methods necessarily result in a tree, since, at each agglomeration step, two clusters have to be merged. This may bias the data analysis process if, for example, a cluster is almost equally attracted by two others. In this paper we propose a new method that allows clusters to overlap until a strong cluster attraction is reached, based on a density criterion. The resulting hierarchical structure, called a quasi-dendrogram, is represented as a directed acyclic graph and combines the advantages of hierarchies with the precision of a less arbitrary clustering. We validate our work with extensive experiments on real data sets and compare it with existing tree-based methods, using a new measure of similarity between heterogeneous hierarchical structures.

1 Introduction

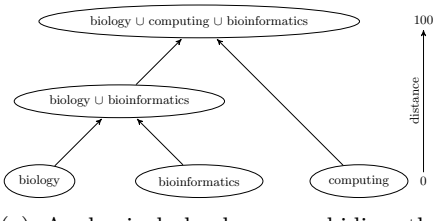
Agglomerative hierarchical clustering methods are widely used to analyze large amounts of data. These successful methods construct a dendrogram – a tree structure – that enables a natural exploration of data which is very suitable even for non-expert users. Various tools offer intuitive top-down or bottom-up exploration strategies, zoom-in and zoom-out operations, etc.

Let us consider the following real-life scenario: a social science researcher would like to understand the structure of specific scientific domains based on a large corpus of publications, such as dblp or Wiley. A contemporary approach is to construct a word embedding [23] of the key terms in publications, that is, to map terms into a high-dimensional space such that terms frequently used in the same context appear close together in this space (for the sake of simplicity, we omit interesting issues such as preprocessing, polysemy, etc.). Identifying for example the denser regions in this space directly leads to insights on the key terms of Science. Moreover, building a dendrogram of key terms using an agglomerative method is typically used [9,14] to organize terms into hierarchies. This dendogram (Figure 1a) eases data exploration and is understandable even for non-specialists of data science.

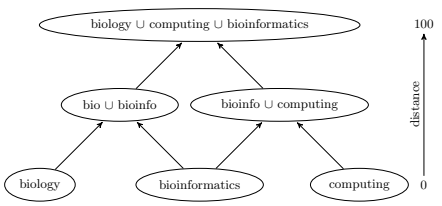
Despite its usefulness, the dendrogram structure might be limiting. Indeed, any embedding of key terms has a limited precision, and key terms proximity is a debatable question. For example, in Figure 1a, we can see that the *bioinformatics*

key term is almost equally attracted by *biology* and *computing*, meaning that these terms appear frequently together, but in different contexts (e.g. different scientific conferences). Unfortunately, with classical agglomerative clustering, a merging decision has to be made, even if the advantage of one cluster on another is very small. Let us suppose that arbitrarily, *biology* and *bioinformatics* are merged. This may suggest to our analyst (not expert in computer science) that *bioinformatics* is part of *biology*, and its link to *computing* may only appear at the root of the dendrogram. Clearly, an interesting part of information is lost in this process.

In this paper, our goal is to combine the advantages of hierarchies while avoiding early cluster merge. Going back to the previous example, we would like to provide two different clusters showing that *bioinformatics* is closed both to *biology* and *computing*. At a larger level of granularity, these clusters will still collapse, showing that these terms belong to a broader community. This way, we deviate from the strict notion of trees, and produce a directed acyclic graph that we call a quasi-dendrogram (Figure 1b).



(a) A classical dendrogram, hiding the early relationship between *bioinformatics* and *computing*.



(b) A **quasi-dendrogram**, preserving the relationships of *bioinformatics*.

Fig. 1: Dendrogram and quasi-dendrogram for the structure of Science.

Our contributions are the following:

- We propose an agglomerative clustering method that produces a directed acyclic graph of clusters instead of a tree, called a quasi-dendrogram,
- We define a density-based merging condition to identify these clusters,
- We introduce a new similarity measure to compare our method with other, quasi-dendrogram or tree-based ones,
- We show through extensive experiments on real and synthetic data that we obtain high quality results with respect to classical hierarchical clustering, with reasonable time and space complexity.

The rest of the paper is organized as follows: Section 2 describes our proposed overlapping hierarchical clustering framework[1]. Section 3 details our experimental evaluation. Section 4 presents the related works, while Section 5 concludes the paper.

[1] Source code available at https://gitlab.inria.fr/ijeantet/ohc

2 Overlapping hierarchical clustering

2.1 Intuition and basic definitions

In a nutshell, our method obtains clusters in a gradual agglomerative fashion and in a precise way. At each step, when we increase the neighbourhood of the clusters by including more interconnections, we consider the points that fall in this connected neighbourhood and we take the decision to merge some of them whenever they are connected enough to a cluster using a *density criterion* λ. Taking interconnections into account may lead to overlapping clusters.

More precisely, we consider a set $V = \{X_1, \ldots, X_N\}$ of N points in a n-dimensional space, i.e. $X_i \in V \subset \mathbb{R}^n$ where $n \geq 1$ and $|V| = N$. In order to explore this space in an iterative way, we consider points that are close up to a limit distance $\delta \geq 0$. We define the δ-neighbourhood graph of V as follows:

Definition 1 (δ-neighbourhood graph). *Let $V \subset \mathbb{R}^n$ be a finite set of data points and $E \subset V^2$ a set of pair of elements of V, let d be a metric on $\mathbb{R}^n$ and let $\delta \geq 0$ be a positive number. The δ-neighbourhood graph $G_\delta(V, E)$ is a graph with vertices labelled with the data points in V, and where there is an edge $(X, Y) \in E$ between $X \in V$ and $Y \in V$ if and only if $d(X, Y) \leq \delta$.*

Property 1. If $\delta = 0$ then the δ-neighbourhood graph consists of isolated points while if $\delta = \delta_{max}$, where δ_{max} is the maximum distance between any two nodes in V then $G_\delta(V, E)$ is the complete graph on V.

Varying δ will allow to progressively extend the neighbourhood of the vectors to form bigger and bigger clusters. Clusters will be formed according to the density of a region of the graph.

Definition 2 (Density). *The density [16] $dens(G)$ of a graph $G(V, E)$ is given by the ratio of the number of edges of G to the number of edges of G if it were a complete graph, that is, $dens(G) = \frac{2|E|}{|V|(|V|-1)}$. If $|V| = 1$, $dens(G) = 1$.*

A cluster is simply defined as a subset of the nodes of the graph and its density is defined as the density of the corresponding subgraph.

2.2 Computing hierarchies with overlaps

Our algorithm, called OHC, computes a hierarchy of clusters that we can identify in the data. We call the generated structure a quasi-dendrogram and it is defined as follows.

Definition 3 (Quasi-dendrogram). *A quasi-dendrogram is a hierarchical structure, represented as a directed acyclic graph, where the nodes are labelled with a set of data points, the clusters, such as:*

- *The leaves (i.e. the nodes with 0 in-degree) correspond to the singletons, i.e. contain a unique data point. The level of the leaf nodes is 0.*

- *There is only one root node (node with 0 out-degree) that corresponds to the set of all the data points.*
- *Each node (except the root node) has one or more parent nodes. The parent relationship corresponds to inclusion of the corresponding clusters.*
- *The nodes at a level δ represent a set of (potentially overlapping) clusters that is a cover of all the data points. Also, for each pair of points of a given cluster, it exists a path between points of this cluster that have a distance less than δ. In other terms, a node contains a part of a connected subgraph of the δ-neighbourhood graph.*

The OHC method works as presented in Algorithm 1. We first compute the distance matrix of the data points (I3). We chose the cosine distance, widely use in NLP. Then we construct and maintain the δ-neighbourhood graph $G_\delta(V, E)$, starting from $\delta = 0$ (I4).

We also initialize the set of clusters, i.e. the leaves of our quasi-dendrogram, with the individual data points (I4). At each iteration, we increase δ (I6) and consider the new added links to the graph (I8) and the impacted clusters (I9). We extend these clusters by integrating the most linked neighbour vertices if the density does not change more than a given threshold λ (I10-15). We remove all the clusters included in these extended clusters (I16) and add the new set of clusters to the hierarchy as a new level (I18). We stop when all the points are in the same cluster which means that we reached the root of the quasi-dendrogram.

Also to improve the efficiency of this algorithm we use dynamic programming to avoid to recompute information related to the clusters like their density and the list of their neighbour vertices. It lead to significant improvements in the execution time of the algorithm. We will discuss this further in the Section 3.3.

Property 2 ($\lambda = 0$). When $\lambda = 0$, each level δ_i of a quasi-dendrogram contains exactly the cliques (complete subgraphs) of the δ_i-neighbourhood graph G_{δ_i}.

Property 3 ($\lambda = 1$). When $\lambda = 1$, each level δ_i of a quasi-dendrogram contains exactly the connected subgraphs of the δ_i-neighbourhood graph G_{δ_i}.

3 Experimental evaluation

3.1 Experimental methodology

Tests: The tests we performed were focused on the quality of the hierarchical structures produced by our algorithm. To measure this quality we used the classical hierarchy produced by *SLINK*, an optimal single-linkage clustering algorithm proposed in *Sibson et al., 1973* [28], as a baseline. Our goal was to study the behaviour of the **merging criterion** parameter λ that we introduced, as long as its influence on the **execution time**, to verify if for $\lambda = 1$ we experimentally obtain the same hierarchy as *SLINK* (Prop. 3) and hence observe the **conservativeness** of our algorithm. We also compared our method to other agglomerative methods such as the *Ward* variant [29] and *HDBSCAN** [8]. To compare such structures we needed to create a new similarity measure which is described in Section 3.2.

Algorithm 1 Overlapping Hierarchical Clustering (OHC)

Input:
- $V = \{x_1, \ldots, x_N\}$, N data points.
- $\lambda \geq 0$, a merging density threshold.

Output: quasi-dendrogram H.
Preprocessing: obtain $\Delta = (\delta_1, \ldots, \delta_m)$ the distances between data points in increasing order.
Initialization:
- Create the graph $G(V, E_0 = \emptyset)$.
- Set a list of clusters $C = [\{x_1\}, \ldots, \{x_N\}]$.
- Add the list of clusters to the level 0 of H.

i=1.
while $\#C > 1$ and $i \leq m$ **do**
  for each pair $(u, v) \in V^2$ such as $d(u, v) = \delta_i$ **do**
    Add (u, v) to $E_{\delta_{i-1}}$.
    Determine the impacted clusters C_{imp} of C containing either u or v.
  for each impacted cluster $C_{imp_j} \in C_{imp}$ **do**
    Look for the points $\{p_1, \ldots, p_k\}$ that are the most linked to C_{imp_j} in G_{δ_i}.
    Compute the density $dens(S_j)$ of the subgraph $S_j = C_{imp_j} \cup \{p_1, \ldots, p_k\}$.
    if $S_j \neq C_{imp_j}$ and $|dens(S_j) - dens(C_{imp_j})| \leq \lambda$ **then**
      Continue to add the most linked neighbors to S_j the same way if possible.
      When S_j stops growing remove C_{imp_j} from the list of clusters C and add S_j to the list of new clusters C_{new}.
  Remove all cluster of C included in one of the clusters of C_{new}.
  Concatenate C_{new} to C.
  Add the list of clusters to the level δ_i of H.
  i=i+1.
return H

Datasets: To partially see the scalability of our algorithm but also to avoid too long running times we had to limit the size of the datasets to few thousand vectors. To be able to compare the results, we run the tests on datasets of same size that we fixed to **1000 vectors**.

- The first dataset is composed of 1000 **randomly** generated 2-dimensional points.
- To test the algorithm on real data and in our motivating scenario, the second dataset was created from the **Wiley** collection via their API[2]. We extracted the titles and abstracts of the scientific papers and trained a word embedding model on the data of a given period of time by using the classical *SGNS* algorithm from *Mikolov et al., 2013* [22] following the recommendation of *Levy et al., 2015* [20]. We set the vocabulary size to only 1000 key words per year even though this dataset allows us to extract up to 50000 of them. This word embedding algorithm created 1000 300-dimensional vectors for each year over 20 years.

Experimental setting: All our experiments are done on a Intel Xeon 5 Core 1.4GHz, running MacOS 10.2 on a SSD hard drive. Our code is developed with Python 3.5 and the visualization part was done on a Jupyter NoteBook. We used the *SLINK* and Ward implementations from the scikit-learn python package and the *HDBSCAN** implementation of *McInnes et al.* [21].

[2] https://onlinelibrary.wiley.com/library-info/resources/text-and-datamining

3.2 A hierarchy similarity measure

As there is no ground truth on the hierarchy of the data we used, we need a similarity measure to compare the hierarchical structures produced by hierarchical clustering algorithms. The goal is not only to compare the topology but also the content of the nodes of the structure. However up to our knowledge there is very little in the literature about hierarchy comparison especially when the structure is similar to a DAG or a quasi-dendrogram. *Fowlkes and Mallows, 1983* [19] defined a similarity measure per level and the new similarity function we propose is based on the same principle. First we construct a similarity between two given levels of the hierarchies, and then we extend it to the global structures by exploring all the existing levels.

Level similarity: Given two hierarchies h_1 and h_2 and a cardinality i, we assume that it is possible to identify a set l_1 (resp. l_2) of i clusters for a given level of hierarchy h_1 (resp. h_2). Then, to measure the similarity between l_1 and l_2, we take the maximal Jaccard similarity among one cluster of l_1 and every clusters of l_2. The average of these similarities, one for each cluster of l_1, will give us the similarity between the two sets. If we consider the similarity matrix of h_1 and h_2 with a cluster of l_1 for each row, a cluster of l_2 for each column and the Jaccard similarity between each pair of clusters at the respective coordinates in the matrix, we can compute the similarity between l_1 and l_2 by taking the average of the maximal value for each row. Hence, the similarity function between two sets of clusters l_1, l_2 is defined as:

$$sim_l(l_1, l_2) = mean\{max\{J(c_1, c_2) \mid c_2 \in l_2\} | c_1 \in l_1\} \tag{1}$$

where J is the Jaccard similarity function.

However, taking the maximal value of each row shows how the clusters of the first set are represented in the second. If we take the maximal value of each column we will see the opposite, i.e. how the second set is represented in the first set. Hence with this definition the similarity might not be symmetrical so we propose this corrected similarity measure that shows how both sets are represented in the other one:

$$sim_l^*(l_1, l_2) = mean(sim_l(l_1, l_2), sim_l(l_2, l_1)) \tag{2}$$

Complete similarity: Now that we can compare two levels of the hierarchical structures, we can simply average the similarity for each corresponding levels of the same size. For classical dendograms, each level has a distinct number of clusters so identification of levels is easy. Conversely, our quasi-dendrograms may have several distinct levels (pseudo-levels) with the same number of clusters. If so, we need to find the best similarity between these pseudo-levels. For a given level (i.e. number of clusters), we want to build a matching M that maps each pseudo-level $l_1^1, l_1^2, ...$ of h_1 to at least one pseudo-level $l_2^1, l_2^2, ...$of h_2 and conversely (see Figure 2). This matching M should maximize the similarity between pseudo-levels while preserving their hierarchical relationship. That is, for

a, b, c, d representing the height of pseudo-levels in the hierarchies, if $(l_1^a, l_2^c) \in M$ and $(l_1^b, l_2^d) \in M$, then $(b \geq a \rightarrow d \geq c)$ or $(b < a \rightarrow d < c)$ (no "crossings" in M, such as $((l_1^{231}, l_2^{303})$ with $(l_1^{230}, l_2^{304}))$.

To produce this mapping, our simple algorithm is the following. We initialize M and two pointers with the two highest pseudo-levels $((l_1^{231}, l_2^{304})$, step 1 of Fig. 2). At each step, for each hierarchy, we consider current pointers and their children, and compute all their similarities (step 2). We then add pseudo-levels with maximal similarity to M (here, $(l_1^{230}, l_2^{303}))$. Whenever a child is chosen, the respective pointer advances, and at each step, at least one pointer advances. Once pseudo-levels have been consumed on one side, ending with l, we can finish the process by adding (l^f, l) to M for all remaining pseudo-level l' on the other side (here, $l = l_1^{230}$. On our example, the final matching is $M = \{(l_1^{231}, l_2^{304}), (l_1^{230}, l_2^{303}), (l_1^{230}, l_2^{302}), (l_1^{230}, l_2^{301}), (l_1^{230}, l_2^{300}), (l_1^{230}, l_2^{299})\}$.

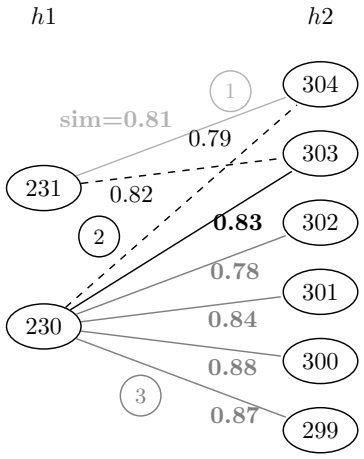


Fig. 2: Computing the similarity between two quasi-dendograms h_1 and h_2 for levels having the same number of clusters.

Finally, from (2) we define the similarity between two hierarchies as

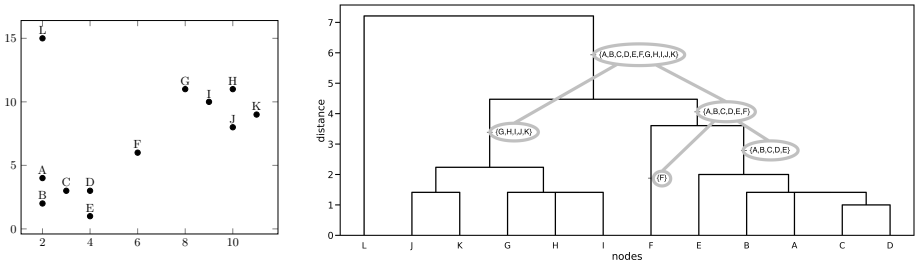
$$sim(h_1, h_2) = mean\{sim_l^*(l_1, l_2) | (l_1, l_2) \in (h_1, h_2) \ \& \ (l_1, l_2) \in M\}. \quad (3)$$

3.3 Experimental results

Expressiveness: With this small following example we would like to present the expressiveness of our algorithm compared to classical hierarchical clustering algorithms such as *SLINK*. On the hand-built example shown in Figure 3a we can clearly distinguish two groups of points, $\{A, B, C, D, E\}$ and $\{G, H, I, J, K\}$ and two points that we can consider as noise, F and L. Due to the chaining effect we expect that the *SLINK* algorithm will regroup the 2 sets of points early in the hierarchy while we would like to prevent it by allowing some cluster overlaps.

Figure 3b shows the dendrogram computed by *SLINK* and we can see as expected that when F merges with the cluster formed by $\{A, B, C, D, E\}$ the next step is to merge this new cluster with $\{G, H, I, J, K\}$.

On the contrary in Figure 4 that presents the hierarchy built with our method for a specific merging criterion, we can see an example of diamond shape that is specific to our quasi-dendrogram. For simplicity the view here slightly differs from the quasi-dendrogram definition as we used dashed arrows to represent the provenance of some elements of a cluster instead of going further down in hierarchy to have a perfect inclusion and respect the lattice-like structure. The



(a) (b)

Fig. 3: A hand-built example (a) and its *SLINK* dendrogram (b).

merge between the clusters $\{A, B, C, D, E\}$ and $\{G, H, I, J, K\}$ is delayed to the very last moment and the point F will belong to these 2 clusters instead of forcing them to merge. Also depending on the merging criterion we obtain different hierarchical structures by merging earlier of later some clusters.

Merging criterion: As we can see in Figure 5b when the merging criterion increases we obtain a hierarchy more and more similar to the one produced by the classical *SLINK* algorithm until we obtain exactly the same for a merging criterion of 1. Knowing this fact it is also normal to have a similarity between OHC and *Ward* (resp. *HDBSCAN**) hierarchies converging to the similarity between *SLINK* and *Ward* (resp. *HDBSCAN**) hierarchies. However we can notice that the OHC and *Ward* hierarchies are the most similar for a merging criterion smaller than 1.

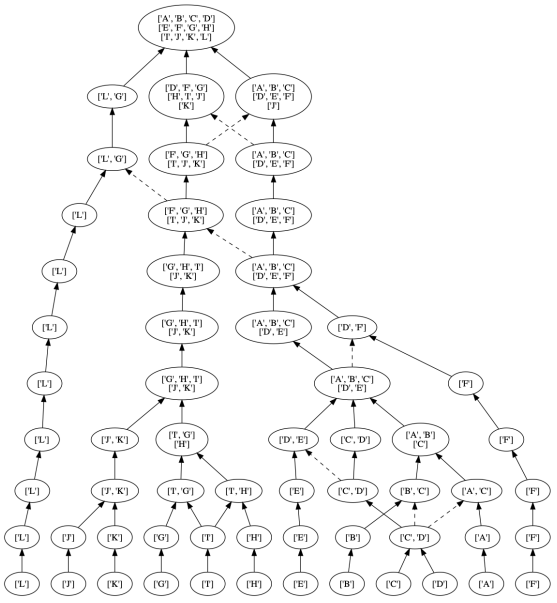
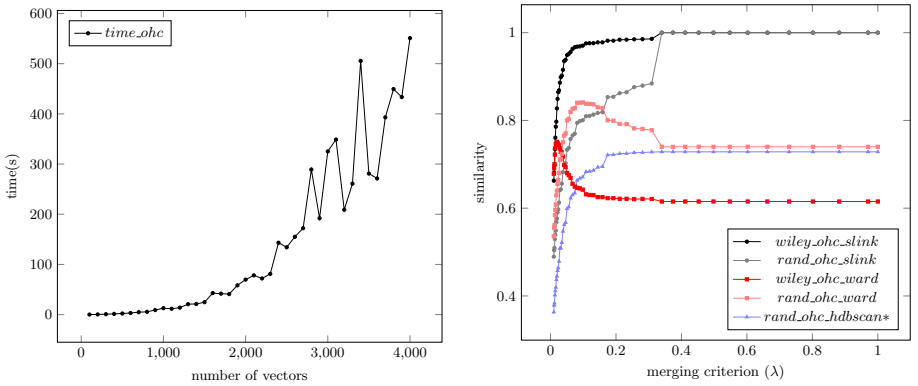


Fig. 4: *OHC* quasi-dendrogram obtained from the hand-built example in Figure 3a for $\lambda = 0.2$.

Execution time: We observe that when the merging criterion increases the execution time decreases. It is due to the fact that when the merging criterion increases we are more likely to completely merge clusters so we reach faster the top of the hierarchy. It means less levels and less overlapping clusters so less computation. However in this case we have the same drawback of chaining effect as the single-



(a) Execution time according to the number of vectors.

(b) Similarity between hierarchical structures according to the merging criterion.

Fig. 5: Study of the merging criterion.

linkage clustering that we wanted to avoid. Even if it was not the objective of this work we set $\lambda = 0.1$, as it is an interesting value according to the study of the merging criterion (5a), to observe the evolution of the execution time (5a). The trend gives a function in $O(n^{2.45})$ so to speed up the process and scale up our algorithm is it possible to precompute a set of possibly overlapping clusters over a given δ-neighbourhood graph with a classical method, for instance CLIQUE, and build the OHC hierarchy on top of that.

4 Related work

Our goal is to group together data points represented as vectors in $\mathbb{R}^n$. For our motivating application domain of understanding the structure of scientific fields, it is important to construct structures (i) that are hierarchical, (ii) that allow overlaps between the identified groups of vectors and (iii) which groups (clusters) are related to dense areas of the data. There are a number of other application domains where obtaining a structure with these properties is important. In the following, we relate our work to relevant literature.

Hierarchical clustering: There exist two kinds of hierarchical clustering. Divisive methods follow a top-down strategy while agglomerative techniques compute the hierarchy in a bottom-up fashion. It produces the well known dendrogram structure [1]. One of the oldest methods is the single-linkage clustering that first appeared in the work of *Florek et al., 1951* [18]. It had many improvements over the years until an optimal algorithm named *SLINK* proposed by *Sibson, 1973* [28]. However the commonly cited drawback of the single-linkage clustering is that it is not robust to noise and suffers from chaining effects (spurious points merging clusters prematurely). It led to the invention of many variants with their advantages and disadvantages. In the NLP world we have for instance the *Brown*

clustering [7] and its generalized version [13]. The drawback of choosing the number of clusters beforehand present in the original Brown clustering is corrected in the generalized version. Researchers also tried to address directly the chaining effect problem with approaches through defining new objective functions such as the *Robust Hierarchical Clustering* [4,11]. However these variants do not allow any overlaps in the clusters. Other variants tried to allow this fuzzy clustering in the hierarchy such as *SOHC* [10], a hierarchical clustering based on a spatial overlapping metric but with a fixed number of clusters, or *HCOSM* [26], that use an overlap similarity measure to merge clusters and then compute a hierarchy from an already determined set of clusters. Generalization of dendrogram to more complex structures like *Pyramidal Clustering* [15] and *Weak Hierarchies* [5] were also proposed. We can find examples to prove that our method produces even more general hierarchical structures that include the weak hierarchies.

Density-based clustering: Another important class of work is the density-based clustering. Here, clusters are defined as regions in the data that have a higher density. The data points in the sparse areas that are required to separate clusters are considered as noise or border points. One of the most widely-used algorithms of this category is *DBSCAN* defined by *Ester et al., 1996* [17]. This method connects data points that satisfy a specific density-based criterion: the minimum number of other data points within a given radius must be above a predefined threshold. The main advantage of this method is that it allows detecting clusters of arbitrary shapes. More recently improved versions of *DBSCAN* were proposed such as *HDBSCAN** [8]. This new variant not only improved notions from *DBSCAN* and *OPTICS* [3] but also proposed a procedure to extract a simplified cluster tree from the reachability relation which allows determining a hierarchy of the clusters but again with no overlapping.

Overlapping clustering: Fuzzy clustering methods [6] allow that certain data points belong to multiple clusters with a different level of confidence. In this way, the boundary of clusters is fuzzy and we can talk about overlaps of these clusters. In our definition it is a different notion, a data point either does or does not belong to a specific cluster and might also belong to multiple clusters. While *HDBSCAN* is closely related to connected components of certain level sets, the clusters do not overlap (since overlap would imply the connectivity.)

Community detection in networks: A number of algorithmic methods have been proposed to identify communities. The first kind of methods produces a partition where a vertex can belong to one and only one community. Following the *modularity* function of *Newman and Girvan* [24], numerous quality functions have been proposed to evaluate the goodness of a partition with a fundamental drawback, the now proved existence of a resolution limit. The second kind of methods, such as *CLIQUE* [2], *k-clique* [25], *DBLC* [31] or *NMF* [30], aims at finding sets of vertices that respect an edge density criterion which allows overlaps but can lead to incomplete cover of the network. Similarly to *HCOSM*, the method *EAGLE* [27] builds a dendrogram over the set of predetermined clusters, here the maximal cliques of the network so overlaps appear only at the leaf level. Coscia et atl. [12] have proposed an algorithm to reconstruct a

hierarchical and overlapping community structure of a network, by hierarchically merging local ego neighbourhoods.

5 Conclusion and future work

We propose an overlapping hierarchical clustering framework. We construct a quasi-dendrogram hierarchical structure to represent the clusters that is however not necessarily a tree (of specific shape) but a directed acyclic graph. In this way, at each level, we represent a set of possibly overlapping clusters. We experimentally evaluated our method using several datasets and also our new similarity measure that hence proved its usefulness. If the clusters present in the data show no overlaps, the obtained clusters are identical to the clusters we can compute using agglomerative clustering methods. In case of overlapping and nested clusters, however, our method results in a richer representation that can contain relevant information about the structure of the clusters of the underlying dataset. As a future work we plan to identify interesting clusters on the basis of the concept of stability. Such methods give promising results in the context of hierarchical density-based clustering [21], but the presences of overlaps in the clusters requires specific considerations.

References

1. Achtert, E.: Hierarchical Subspace Clustering. Ph.D. thesis, lmu (2007)
2. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic subspace clustering of high dimensional data. Data Mining and Knowledge Discovery **11**(1), 5–33 (2005)
3. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: Optics: ordering points to identify the clustering structure. In: ACM Sigmod record. vol. 28, pp. 49–60. ACM (1999)
4. Balcan, M.F., Liang, Y., Gupta, P.: Robust hierarchical clustering. The Journal of Machine Learning Research **15**(1), 3831–3871 (2014)
5. Bandelt, H.J., Dress, A.W.: Weak hierarchies associated with similarity measures—an additive clustering technique. Bulletin of mathematical biology **51**(1), 133–166 (1989)
6. Bezdek, J.C.: Pattern recognition with fuzzy objective function algorithms. Springer Science & Business Media (2013)
7. Brown, P.F., Desouza, P.V., Mercer, R.L., Pietra, V.J.D., Lai, J.C.: Class-based n-gram models of natural language. Computational linguistics **18**(4), 467–479 (1992)
8. Campello, R.J., Moulavi, D., Zimek, A., Sander, J.: Hierarchical density estimates for data clustering, visualization, and outlier detection. ACM Transactions on Knowledge Discovery from Data (TKDD) **10**(1), 5 (2015)
9. Chavalarias, D., Cointet, J.P.: Phylomemetic patterns in science evolution - the rise and fall of scientific fields. PloS one **8**(2), e54847 (2013)
10. Chen, H., Guo, G., Huang, Y., Huang, T.: A spatial overlapping based similarity measure applied to hierarchical clustering. In: Fuzzy Systems and Knowledge Discovery. FSKD'08. vol. 2, pp. 371–375. IEEE (2008)
11. Cohen-Addad, V., Kanade, V., Mallmann-Trenn, F., Mathieu, C.: Hierarchical clustering: Objective functions and algorithms. In: Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 378–397. SIAM (2018)

12. Coscia, M., Rossetti, G., Giannotti, F., Pedreschi, D.: Uncovering hierarchical and overlapping communities with a local-first approach. ACM Trans. Knowl. Discov. Data **9**(1), 6:1–6:27 (Aug 2014)
13. Derczynski, L., Chester, S.: Generalised brown clustering and roll-up feature generation. In: AAAI. pp. 1533–1539 (2016)
14. Dias, L., Gerlach, M., Scharloth, J., Altmann, E.G.: Using text analysis to quantify the similarity and evolution of scientific disciplines. Royal Society open science **5**(1), 171545 (2018)
15. Diday, E.: Une représentation visuelle des classes empiétantes: les pyramides (1984)
16. Diestel, R.: Graph theory. 2005. Grad. Texts in Math **101** (2005)
17. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd. vol. 96, pp. 226–231 (1996)
18. Florek, K., Łukaszewicz, J., Perkal, J., Steinhaus, H., Zubrzycki, S.: Sur la liaison et la division des points d'un ensemble fini. In: Colloquium Mathematicae. vol. 2, pp. 282–285 (1951)
19. Fowlkes, E.B., Mallows, C.L.: A method for comparing two hierarchical clusterings. Journal of the American statistical association **78**(383), 553–569 (1983)
20. Levy, O., Goldberg, Y., Dagan, I.: Improving distributional similarity with lessons learned from word embeddings. Transactions of the Association for Computational Linguistics **3**, 211–225 (2015)
21. McInnes, L., Healy, J.: Accelerated hierarchical density based clustering. In: Data Mining Workshops (ICDMW), 2017 IEEE International Conference on. pp. 33–42. IEEE (2017)
22. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
23. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
24. Newman, M.E., Girvan, M.: Finding and evaluating community structure in networks. Physical review E **69**(2), 026113 (2004)
25. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. nature **435**(7043), 814 (2005)
26. Qu, J., Jiang, Q., Weng, F., Hong, Z.: A hierarchical clustering based on overlap similarity measure. In: Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Eighth ACIS International Conference on. vol. 3, pp. 905–910. IEEE (2007)
27. Shen, H., Cheng, X., Cai, K., Hu, M.B.: Detect overlapping and hierarchical community structure in networks. Physica A: Statistical Mechanics and its Applications **388**(8), 1706–1712 (2009)
28. Sibson, R.: Slink: an optimally efficient algorithm for the single-link cluster method. The computer journal **16**(1), 30–34 (1973)
29. Ward Jr, J.H.: Hierarchical grouping to optimize an objective function. Journal of the American statistical association **58**(301), 236–244 (1963)
30. Yang, J., Leskovec, J.: Overlapping community detection at scale: a nonnegative matrix factorization approach. In: Proceedings of the sixth ACM international conference on Web search and data mining. pp. 587–596. ACM (2013)
31. Zhou, X., Liu, Y., Wang, J., Li, C.: A density based link clustering algorithm for overlapping community detection in networks. Physica A: Statistical Mechanics and its Applications **486**, 65–78 (2017)