Benefits of hypergraphs for density-based clustering

Louis Hauseux, Konstantin Avrachenkov, Josiane Zerubia

Benefits of hypergraphs for density-based clustering

Louis HAUSEUX
Inria, Université Côte d'Azur
NEO *team* & AYANA *team*
Sophia Antipolis, France
louis.hauseux@inria.fr

Konstantin AVRACHENKOV
Inria, Université Côte d'Azur
NEO *team*
Sophia Antipolis, France
konstantin.avratchenkov@inria.fr

Josiane ZERUBIA
Inria, Université Côte d'Azur
AYANA *team*
Sophia Antipolis, France
josiane.zerubia@inria.fr

***Abstract*—Many of clustering algorithms are based on density estimates in $\mathbb{R}^d$. Building *geometric graphs* on the dataset $\mathcal{X}$ is an elegant way of doing this. In fact, the connected components of a geometric graph match exactly with the *high-density clusters* of the 1-Nearest Neighbor density estimator.**

In this paper, We show that the natural way to generalize geometric graphs is to use *hypergraphs* with a more restrictive notion of connected component called K-Polyhedron. Herein, we prove that K-polyhedra correspond to high-density clusters of K-Nearest Neighbors density estimator. Furthermore, the *percolation* phenomenon is omnipresent behind the family of clustering algorithms we look at in this paper.

***Index Terms*—hierarchical clustering, density estimator, geometric graphs, hypergraphs, percolation**

I. INTRODUCTION

Cluster analysis – or *clustering* – involves the process of categorizing a collection of items in a manner where items within the same group, called a *cluster*, exhibit greater similarity to one another compared to those in different groups.

When the dataset $\mathcal{X} \subset \mathbb{R}^d$ is made of points in the Euclidean space, considering the point generation density f and its *high-density clusters* is a natural way of tackling this clustering problem [1] (see Fig. 1).

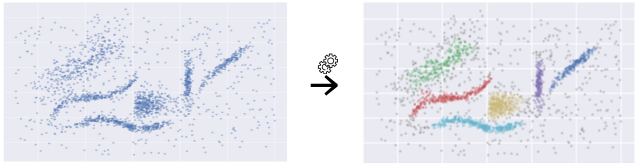

Fig. 1. Result of the Hierarchical Density-Based SCAN (HDBSCAN) algorithm [2] on a 2D toy dataset [3].

Precise discretization of the Eucliden space $\mathbb{R}^d$ is often intractable (and brings its own set of technical and theoretical problems).

One classical solution consists in constructing a graph whose nodes are the points of $\mathcal{X}$ and whose edges connect nearby points. Looking at *geometric graphs* [4] allows us to find the *high-density clusters* of the 1-Nearest Neighbor estimator [5] without constructing a density estimator $\hat{f}$ on the whole space.

To gain in robustness, it is natural to try to replace this 1-Nearest Neighbor by a K-Nearest Neighbors density estimator [5]. This is the key-idea of algorithms such as the *Robust Single-Linkage* [6] or HDBSCAN [2], [3].

However, anticipating explanations in Fig. 4, we note that these algorithms introduce a strong constraint on the vertices while still having relaxed constraints on edges.

We illustrate this theoretical weakness with two practical cases: A toy dataset of the © Scikit-Learn's Clustering webpage [7] and studying the small clusters of an olive oil dataset [8], [9].

To tackle this issue, we propose a novel method using *hypergraphs* rather than standard graphs. Hypergraphs, with edges comprising more than two nodes, allow us to work with more constrained notions of connectivity and therefore more restrictive connected components called K-Polyhedra (see Definition in Section III-A). We show that the K-Polyhedra match with the high-density clusters of the density estimator $\hat{f}_{K\text{-NN}}$ (see Theorem in Section III-C).

II. CLASSICAL MATHEMATICAL MODEL

Assume that the dataset $\mathcal{X}_n := \{x_1, \dots, x_n\} \subset \mathbb{R}^d$ is a cloud of n points plotted IID according to a probability measure with density $f : \mathbb{R}^d \to \mathbb{R}_+$.

Therefore, the underlying structure of the point cloud $\mathcal{X}_n$ lies entirely in this density function f.

With the very intuitive idea that the different clusters are represented by the "peaks" [10] of density, HARTIGAN [11] defined the high-density clusters $H_f(r)$ at level r as the different connected components of the level set L_r

$$L_r := \left\{x \in \mathbb{R}^d \ : \ f(x) \geq r\right\}.$$

By varying the level r, we can obtain a *hierarchical clustering* (*cf.* Fig. 2). Given a cluster C, *i.e.* a connected component of L_r, the 'discrete' cluster on $\mathcal{X}_n$ is then defined by

$$C^{\text{discrete}} := C \cap \mathcal{X}_n.$$

The hierarchical clustering can be represented by a tree, the *dendrogram*. See ROLLE & SCOCCOLA [12] for a much more extensive presentation of hierarchical clustering.

The first author would like to thank the Université Côte d'Azur (UCA) DS4H Investments in the Future project managed by the National Research Agency (ANR, reference number ANR-17-EURE-0004) and 3IA Côte d'Azur for partial funding of his PhD thesis. All the authors acknowledge a partial support by Nokia Bell Labs "Distributed Learning and Control for Network Analysis" and Bpifrance in collaboration with Airbus D&S (LiChIE contract, 2020-2024).

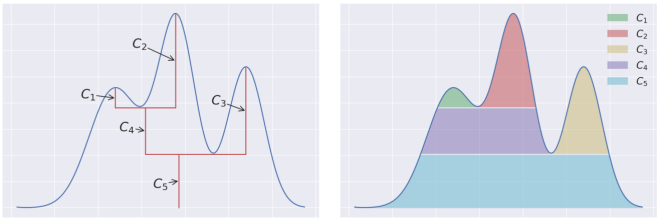

Fig. 2. Hierarchical clustering. Each cluster is associated to its *relative excess of mass* [3], that is the area of the coloured zone. © Images taken from [3]

A. Two approaches for density-based clustering.

First, if there is an estimator $\hat{f}$ of the density f, it is then possible to estimate $H_f(r)$ with $H_{\hat{f}}(r)$. This solution might be adapted for Euclidean spaces of small dimensions (*e.g.* in $\mathbb{R}^d$ with $d = 1$, 2 or 3)... But it becomes intractable in large dimension: discretization of the space becomes too costly.

A way to bypass this difficulty is to construct a *geometric graph* on the data whose connected components are a good estimator of the discrete clusters. Given a point cloud $\mathcal{X} \subset E$ in the Euclidean space $\mathbb{R}^d$, the geometric graph $\mathcal{G}(\mathcal{X}, r)$ of radius r is the graph whose nodes are the points $x \in \mathcal{X}$ and there is an edge between $x, y \in \mathcal{X}$ if $||x-y|| \leq r$, where $||\cdot||$ denotes the Euclidean distance (see PENROSE [4]).

B. Single-Linkage $\simeq$ Geometric graphs $\simeq$ $\hat{f}_{1\text{-Nearest Neighbor}}$

Single-Linkage algorithm constructs a hierarchical clustering as follows: It starts with the trivial initial clustering (n points for n clusters) $\mathcal{C}_0 = \{C_1^0, \ldots, C_n^0\}$ with $C_i^0 = \{x_i\}$. At each step, we merge the two clusters that are the closest for the distance: $d_{Clust}(C, C') = \min_{x \in C, y \in C'} ||x - y||$.

At step t, the resulting clustering $\mathcal{C}_t = \{C_1^t, C_2^t, \ldots, C_{n-t}^t\}$ corresponds to the connected components of a geometric graph $\mathcal{G}(\mathcal{X}_n, r_t)$ built on $\mathcal{X}_n$. Therefore, Single-Linkage performs *persistent analysis* on geometric graphs $\mathcal{G}(\mathcal{X}_n, r)$.

Furthermore, we propose a new clustering algorithm, called 'Hypergraph-Percol', combining the use of these K-Polyhedra with percolation phenomenon.

See Fig. 3 for an illustration. Density is constant on each half-rectangle, the left one A and the right one B, and is larger on A. We observe on the dendrogram the first percolation phase (for the plateus A). Suddenly, for $r \lesssim 0.07$, plenty of clusters merge. The associated geometric graph $\mathcal{G}(\mathcal{X}_{300}, 0.07)$ has a giant component almost corresponding to $A \cap \mathcal{X}_{300}$.

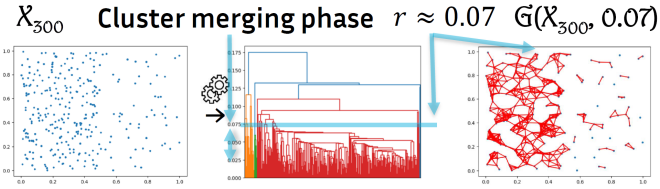

Fig. 3. From left to right: 1) The point cloud $\mathcal{X}_{300}$. 2) The dendogram of the Single-Linkage applied on $\mathcal{X}_{300}$. 3) The geometric graph $\mathcal{G}(\mathcal{X}_{300}, 0.07)$.

HARTIGAN [13] showed that the Single-Linkage algorithm is a consistent estimator of high-density clusters in dimension $d = 1$, but only *fractionnally*[1] consistent in dimension $d \geq 2$.

C. Robust Single-Linkage

To gain in robustness, it is natural to try to replace this 1-Nearest Neighbor by a K-Nearest Neighbors density estimator. CHAUDHURI & DASGUPTA [6] proposed a robust version of the Single-Linkage based on the consistency of the K-Nearest Neighbors density estimator [5].

In the K-Robust version, only points $x \in \mathcal{X}_n$ having more than K neighbors in their r-neighbourhood are considered:

$$\mathcal{X}_n^{K,r} := \{x \in \mathcal{X}_n \ : \ |B(x, r) \cap \mathcal{X}_n| \geq K\} \subseteq \mathcal{X}_n.$$

Then a geometric graph $\mathcal{G}(\mathcal{X}_n^{K,r}, r)$ is constructed on $\mathcal{X}_n^{K,r}$ and the rest of the algorithm is the same as for Single-Linkage. (H)DBSCAN and other algorithms work in the same way [12].

We emphasize that there is now a hiatus between the strong constraint on the points (having K neighbors to appear) and the relaxed condition on edges (once two vertices $x, y \in \mathcal{X}_n^{K,r}$ appear, they are linked with the weak condition $||x-y|| \leq r$).

D. Limitation of the Robust Single-Linkage

1) In theory: For $K > 2$, it is very important to note that the discrete clusters obtained *via* the high-density clusters of the K-Nearest Neighbors density estimator $\hat{f}_{K\text{-NN}}$ are quite different from those of the Robust Single-Linkage (or (H)DBSCAN).

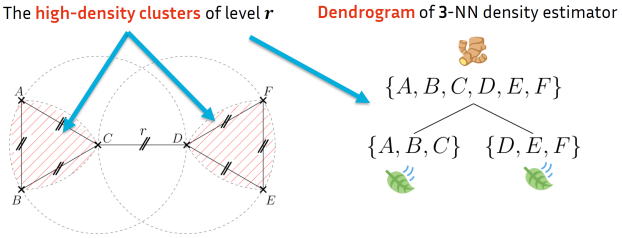

Fig. 4. Left: A cloud $\mathcal{X}_6$ of six points: two equilateral triangle at equidistance r. In red, the two clusters of $H_{\hat{f}_{3\text{-NN}}}$ for level r. Right: The resulting dendrogram of the discrete hierarchical clustering $H_{\hat{f}_{3\text{-NN}}}$.

Look at the Fig. 4 for such an example. The discrete high-density clusters at level r of the point cloud $\mathcal{X}_6$ are composed of the two triangles $\{A, B, C\}$ and $\{D, E, F\}$. Then, for larger level $r' > r$, these two triangles will merge. The hierarchical clustering $H_{\hat{f}_{3\text{-NN}}}$ can thus be represented by the tree on the right-hand side. Whereas the Robust Single-Linkage algorithm merges the two triangles once they appear because the edge $\{C, D\}$ appears at the same time... Consequently, the dendrogram of 3-Robust Single Linkage is also reduced to an unique root–leaf, $\{A, B, C, D, E, F\}$.

[1] A question then arises: How can we measure this recoverable 'fraction'? This led us to define a *percolation rate* in [14]. There is still a lot more to say about it. The phenomenon of *percolation* is omnipresent behind this family of algorithms. Its study is necessary to understand, analyse and compare their performances. To deal with it herein would take us too far... For *continuum percolation*, see MEESTER & ROY [15] or PENROSE [4]; for discrete percolation, the reader can refer to GRIMMETT [16].

2) In practice: This weakness is not purely theoretical. On Scikit-Learn's Clustering webpage [7], HDBSCAN – the algorithm with the best visual results – makes a mistake on a 3-blob example (see Fig. 5): It merges the 3 clusters into 2 clusters. A short chain of points is enough to merge the two vertically aligned clusters.

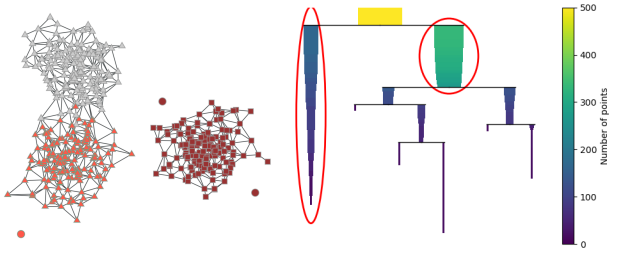

Fig. 5. Left: Result of HDBSCAN on the 3 blobs [7]. Colours are the 3 ground-truth clusters, Shapes are the clustering: only 2 clusters (triangles and squares). Disks are unclustered points. Right: The *condensed clustering tree* with the choice of clusters according to the *excess of mass* criterion [3].

Another example on a real dataset: The Italian olive oil [8], [9], [12], [17]. This dataset was first presented in 1983 [8]. It consists of 572 samples of chemical oil composition, these samples coming from nine different italian regions grouped in three macro-areas: 1) South: North Apulia, South Apulia, Calabria and Sicilia; 2) Sardinia: Inland and Coast; 3) Centro-settentrionale: East Liguria, West Liguria and Umbria.

HDBSCAN is able to recover the 3 geographical 'ground-truth' macro-areas[2] but not the 9 finer region-clusters.

This weakness leads us to look at more general objects than graphs, with more restrictive notions of connectivity.

III. Proposed approach

A. Hypergraphs and K-Polyhedra

Like a graph, a hypergraph is defined by a set of vertices $\mathcal{X}$ and a set of edges E. The difference is that the edges $e \in E$ are not only *pairs* $\{x, y\}$ of vertices $x, y \in \mathcal{X}$ but can be any non-empty subsets of the vertices $e \subseteq \mathcal{X}$. An edge can thus express more elaborated neighbourhood relationships. We will look at a sub-family of hypergraphs called *simplicial complexes* in algebraic topology (see Munkres [19]).

Definition. *A polyhedron of dimension K is defined inductively (see Fig. 6 for an illustration):*

— The convex hull of a hyperedge $e = \{x_{i_0}, \dots, x_{i_K}\}$ of dimension K (with $K+1$ vertices) is a polyhedron of dimension K.

— If two polyhedra of dimension K share a common facet (hyperedge of dimension $K-1$), then their union is still a polyhedron of dimension K.

[2]The geographical location is in fact an important criterion... It should not be the only one. *E.g.* North-Apulia forms a cluster clearly distinct from South-Apulia (and also from the other clusters; look at Fig. 8: it is the last small remaining cluster when everything else has merged). This certainly has more to do with the way of oil fabrication and olive harvesting (either picked still on the tree or picked after they fall) than with the geographical, geological or climatic differences. At the time of the study (1983), some olive farmers still used millenia-old practises (see an article on olive oil in Apulia [18]).

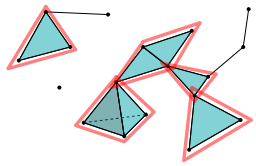

Fig. 6. A hypergraph and its five 2-Polyhedra (or "Triangle-connected" components) surrounded in red.

This notion of connected component on hypergraphs was defined by Atkin [20] for simplicial complexes and called 'q-connectivity'. See [21] for more recent developments.

B. Čech simplicial complexes

The Čech complex $\check{C}(\mathcal{X}_n, r)$ is the hypergraph whose K-dimensional hyperedges $e = \{x_{i_0}, \dots, x_{i_k}\}$ are in $\check{C}(\mathcal{X}_n, r)$ if their *epicentrum* $E(e)$ is not empty:

$$E(e) := \bigcap_{j=0}^{k} \overline{B}\left(x_{i_j}, r/2\right) \neq \emptyset.$$

Note that the 1-skeleton of $\check{C}(\mathcal{X}_n, r)$ corresponds to the geometric graph $\mathcal{G}(\mathcal{X}_n, r)$.

C. Correspondence between K-Polyhedra of $\check{C}(\mathcal{X}_n, r)$ and the high-density clusters of $\hat{f}_{K\text{-NN}}$

Let $\mathcal{X}_n \subset \mathbb{R}^d$ be a point cloud IID generated by a continuous density function f. Let L_r^K be the r-level set of the K-Nearest Neighbors density estimator

$$L_r^K = \left\{x \in \mathbb{R}^d \ : \ |\overline{B}(x, r/2) \cap \mathcal{X}_n| \geq K\right\} = \bigcup_{\substack{e \in \check{C}(\mathcal{X}_n, r) \\ |e| = K}} E(e)$$

and $H_{\hat{f}_{K\text{-NN}}}(r)$ the associated *high-density clusters*, *i.e.* the set of connected components of L_r^K.

Note that for fixed radius r, $L_r^{K+1} \subset L_r^K$, we can therefore 'link' the $(K+1)$-clusters $C \in H_{\hat{f}_{(K+1)\text{-NN}}}(r)$ which are in the same K-cluster $C' \in H_{\hat{f}_{K\text{-NN}}}(r)$. Let us define

$$Link_r^K = \left\{ \bigcup_{\substack{C \in H_{\hat{f}_{(K+1)\text{-NN}}}(r) \\ C \subset C'}} C \ : \ C' \in H_{\hat{f}_{K\text{-NN}}}(r) \right\} \setminus \{\emptyset\}.$$

We define the ε-*dilatation* of the set $X \subset \mathbb{R}^d$ by

$$\delta_\varepsilon(X) = \left\{x \in \mathbb{R}^d \ : \ \overline{B}(x, \varepsilon) \cap X \neq \emptyset\right\}.$$

Theorem. *Let K be the dimension of the polyhedra, r a radius, $\check{C}(\mathcal{X}_n, r)$ the Čech complex of radius r on the cloud $\mathcal{X}_n$ and $V_1, \dots, V_\alpha$ the sets of vertices of its α K-Polyhedra. Then, $\alpha = |Link_r^K|$. Moreover, the V_is are exactly given by*

$$\delta_{r/2}(C) \cap \mathcal{X}_n \quad \text{for} \quad C \in Link_r^K.$$

By looking at the K-Polyhedra, we can therefore identify the clusters of $H_{\hat{f}_{(K+1)\text{-NN}}}(r)$ that are connected in L_r^K. While Robust Single-Linkage (or HDBSCAN) merges all the clusters

of $H_{\hat{f}_{(K+1)\text{-NN}}}(r)$ that were connected after an r-dilatation, which is much less restrictive.

PROOF. *The proof is not very difficult but would take too much space here. The idea is that connecting K-hyperedges of $\check{C}(\mathcal{X}_n, r)$ by means of $(K-1)$-hyperedges is equivalent (apart from $r/2$-dilatation) to connecting $(K+1)$-high-density clusters within K-high-density clusters. The full proof will be submitted to a journal.*

D. The novel 'Hypergraph-Percol' algorithm

Alg. 1 schematically presents how our Hypergraph-Percol works. Varying a radius r, it produces a *persistent analysis* on Čech complexes $\check{C}(\mathcal{X}, r)$. In the spirit of working only after percolation phases, a 'percolation threshold' parameter is introduced, pruning the resulting hierarchical tree.

Algorithm 1 Hypergraph-Percol

Input: $\mathcal{X}$ the point cloud, $K \in \mathbb{N}$ and $PercolThreshold \in \mathbb{N}$
Output: Hierarchical clustering $\hat{H} : r \mapsto \hat{H}(r)$

```
for r from 0 to +∞ do
    Hypergraph ← Č(X, r)
    Polyhedra ← K-Polyhedra(Hypergraph)
    for Polyhedron in Polyhedra do
        if Length(Polyhedron) < PercolThreshold then
            Remove(Polyhedron) from Polyhedra
        end if
    end for
    Ĥ(r) ← Polyhedra
end for
Return Ĥ
```

Once a hierarchical clustering[3] $\hat{H} : r \mapsto \hat{H}(r)$ is obtained, we can compute its *linkage matrix* [22] and draw the associated *condensed clustering tree* [3].

The *relative excess of mass* criterion [3] is then applied to extract the relevant clusters. Multi-clustered points are removed from the final clustering. To obtain an exhaustive clustering, we associate to each unclustered point the cluster of its Nearest Neighbor (within the clustered points).

IV. Experimental results

See Section II-D2 for a presentation of the two datasets.

A. 2D toy dataset of the © Scikit-Learn's Clustering webpage [7]

$\mathcal{X}_{500}$ is composed of 3 blobs (see Fig. 5 and 7). The computation of the K-Polyhedra on the Čech complex $\check{C}(\mathcal{X}_{500}, r)$ is made with the same parameters as in HBDSCAN: $K \leftarrow 2$ and the percolation threshold $PercolThreshold \leftarrow 15$. We also apply the same *relative excess of mass* [3] criterion for the choice of clusters (see the red ellipses in Fig. 7: the *excess of mass* is the area of the vertical bars in the dendrogram). The 2-Polyhedra being a more constraining notion of connected components than the 3-Neighbors of HDBSCAN, the result is less smooth and smaller components appear and have a longer lifespan. See the resulting clustering on Fig. 7. We identify well the *three* clusters (the three shapes; contrary to HDBSCAN, *cf.* Fig. 5). On the cloud $\mathcal{X}_{500}$ made of 500 points, 29 are unclustered, 1 is bi-clustered and 3 are misclustered. Among the 30 bi- or un-clustered points, 28 would be rightly clustered using the 1-Nearest (clustered) Neighbor. Thus, the exhaustive 'Hypergraph-Percol' method – which is totally unsupervized – gives 495 well-clustered points; result similar to classical methods requiring the number of clusters such as Mini-batch K-Means [23] (495), Spectral clustering [24] (494) or Gaussian mixture [25] (494) (see Tab. I).

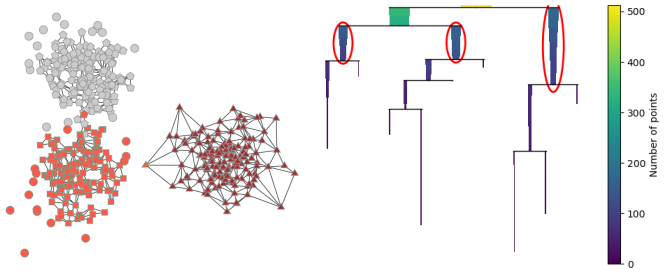

Fig. 7. Left: Results of our algorithm on the 3-blobs. Colours are the 3 ground-truth clusters: gray, orange and red. © Scikit-Learn's Clustering webpage. Shapes are the final clustering using the *condensed clustering tree* [3] (Right). We recover the 3 clusters (triangles, squares and pentagons). Disks are unlabelled points.

TABLE I
RESULTS OF DIFFERENT CLUSTERING ALGORITHMS ON 3 BLOBS [7].

	Methods requiring n_clusters			**Unsupervized methods**	
	K-Means	Spectr. C.	Gauss. M.	HDBSCAN	Hypergraph-Percol
$n\_clusters$	Given	Given	Given	2	**3**
Accuracy	**495/500**	494/500	494/500	X	**495/500**
Rand Index	**0.987**	0.984	0.984	X	**0.987**

B. The Olive Oil dataset [8], [9]

HDBSCAN is able to recover the three macro-areas but not the finer regions. Whereas our 'Hypergraph-Percol' method is. With $K \leftarrow 6$ and $PercolThreshold \leftarrow 30$ we obtain the condensed tree drawn in Fig. 8 with height clusters. Height and not nine because – as with the 'Persistable algorithm' in [12] – the 'Sicilia' cluster (4^{th} in Tab. II) does not appear.

Note that a point may appear in several different clusters. That is why the "number of points" in Fig. 8 is higher than 572, the number of samples. In Tab. II, only points that appear in a single cluster are classified. 394 points are thus clustered with a relative accuracy of $374/394 \approx 94.9\%$. It can be noticed that there are few false positives (apart the diagonal, majority of cells have 0-value – in light gray). Using the Nearest-Neighbor clustered point to obtain an exhaustive clustering, the overall accuracy is $506/572 \approx 88.5\%$ (against $499/572 \approx 87.2\%$ in [12]).

[3]Note that, here, a 'clustering' $H(r)$ may not be a partition of $\mathcal{X}$: some points at the frontier of K-Polyhedra may appear in several clusters.

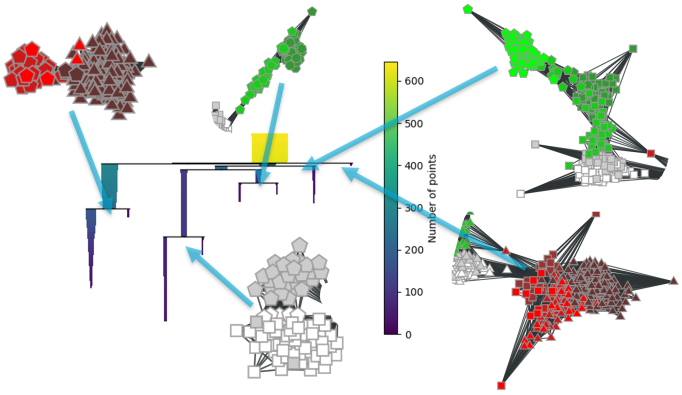

Fig. 8. *Condensed clustering tree* [3] obtained with our Hypergraph-Percol method on the ℓ^2-normalized olive oil dataset [8], [9]. We recover 8 of the 9 geographical 'ground-truth' region clusters. The resulting clustering is given in Tab. II.

TABLE II
RESULTS OF THE PERSISTABLE [12] AND OUR HYPERGRAPH-PERCOL ALGORITHMS ON THE OLIVE OIL DATASET.

Pers./Ours	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**Miss.**
N. Apulia	12/**14**	0/0	0/0	-	0/0	0/0	0/0	0/0	0/0	13/**11**
Calabria	0/0	7/**28**	**1**/1	-	0/0	0/0	0/0	0/0	0/0	48/**27**
S. Apulia	0/0	0/0	100/**167**	-	0/0	0/0	0/0	0/0	0/0	106/**39**
Sicilia	**3**/5	**0**/1	**0**/2	-	0/0	0/0	0/0	0/0	0/0	**33**/28
Inl. Sard.	0/0	0/0	0/0	-	51/**52**	0/0	0/0	0/0	0/0	14/**13**
Coast S.	0/0	0/0	0/0	-	**0**/2	19/**27**	0/0	0/0	0/0	14/**4**
E. Ligur.	0/0	0/0	0/0	-	0/0	0/0	14/**20**	**1**/3	0/0	35/**27**
W. Ligur.	0/0	0/0	0/0	-	0/0	0/0	0/0	29/**41**	0/0	21/**9**
Umbria	0/0	0/0	0/0	-	0/0	0/0	**0**/6	0/0	**42**/25	**9**/20

V. CONCLUSION AND PERSPECTIVES

This paper presents a new clustering method called 'Hypergraph-Percol' for data $\mathcal{X} \subset \mathbb{R}^d$ in the Euclidean space.

We construct hypergraphs (Čech complexes) on $\mathcal{X}$ and use a restrained notion of connectivity called the K-Polyhedron connectivity (Definition III-A). We show that our 'Hypergraph-Percol' clustering technique is theoretically better able to recover *high-density clusters* (Theorem III-C) than other classical density-based clustering algorithms such as *Robust Single-Linkage* [6] or HDBSCAN [2], [3].

This capability is illustrated on two datasets. First, a 2D toy dataset [7] on which HDBSCAN defaulted and recovered only 2 of the 3 clusters while our method works well. Second, an olive oil dataset [8], [9] with geographical clusters as ground-truth. Here, HDBSCAN is not able to detect precisely finer clusters. Again, our method is more robust than HDBSCAN and manages to detect smaller clusters with good accuracy.

Our algorithm provides only a partial clustering, which we completed for unclustered points with a Nearest-Neighbor method.

In the future, it would be interesting to develop a more comprehensive and robust framework than this cobbled-together approach to obtain an exhaustive clustering and test the new method on a larger number of datasets.

In addition, it might be interesting to add a parameter for the cluster selection criterion. We could then automatically obtain a clustering with micro-clusters or macro-clusters depending on the value of this parameter.

REFERENCES

[1] R. J. G. B. Campello, P. Kröger, J. Sander, and A. Zimek, "Density-based clustering," *WIREs Data Mining and Knowledge Discovery*, vol. 10, no. 2, p. e1343, 2020.
[2] R. J. G. B. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Advances in Knowledge Discovery and Data Mining*. Berlin, Heidelberg: Springer, 2013, pp. 160–172.
[3] L. McInnes and J. Healy, "Accelerated hierarchical density based clustering," in *IEEE International Conference on Data Mining Workshops (ICDMW)*, 2017, pp. 33–42.
[4] M. Penrose, *Random Geometric Graphs*. Oxford University Press, 2003, vol. 5.
[5] G. Biau and L. Devroye, *Lectures on the Nearest Neighbor Method*. Springer, 2015, vol. 246.
[6] K. Chaudhuri and S. Dasgupta, "Rates of convergence for the cluster tree," in *Advances in Neural Information Processing Systems*, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds., vol. 23. Curran Associates, Inc., 2010.
[7] Scikit-Learn Library, "Comparing different clustering algorithms on toy datasets," https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html, [Online; accessed 01-February-2024].
[8] M. Forina, C. Armanino, S. Lanteri, and E. Tiscornia, "Classification of olive oils from their fatty acid composition," in *Food research and data analysis: proceedings from the IUFoST Symposium, September 20-23, 1982, Oslo*. London: Applied Science Publishers, 1983.
[9] H. Wickham, D. Cook, H. Hofmann, and A. Buja, "tourr: An r package for exploring multivariate data with projections," *Journal of Statistical Software*, vol. 40, no. 2, p. 1–18, 2011.
[10] J. Tobin and M. Zhang, "A theoretical analysis of density peaks clustering and the component-wise peak-finding algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 2, pp. 1109–1120, Feb 2024.
[11] J. A. Hartigan, *Clustering Algorithms*. John Wiley & Sons, Inc., 1975.
[12] A. Rolle and L. Scoccola, "Stable and consistent density-based clustering," 2023, arXiv.v3.
[13] J. A. Hartigan, "Consistency of single linkage for high-density clusters," *J. of the Am. Stat. Ass.*, vol. 76, no. 374, pp. 388–394, 1981.
[14] L. Hauseux, K. Avrachenkov, and J. Zerubia, "Graph based approach for galaxy filament extraction," in *Complex Networks & Their Applications XII*. Springer, 2024, pp. 384–396.
[15] R. Meester and R. Roy, *Continuum Percolation*, ser. Cambridge Tracts in Mathematics. Cambridge University Press, 1996.
[16] G. Grimmett, *Percolation*. Springer, 1999.
[17] S. R. S. D. Scaldelai, L. C. Matioli and M. Kleina, "MulticlusterKDE: a new algorithm for clustering based on multivariate kernel density estimation," *Journal of Applied Statistics*, vol. 49, no. 1, pp. 98–121, 2022.
[18] C. Ipsen, "Xylella fastidiosa and the Olive Oil Crisis in Puglia," *Gastronomica*, vol. 20, no. 2, pp. 55–66, 05 2020.
[19] J. R. Munkres, *Elements of Algebraic Topology*. CRC Press, 1984.
[20] R. Atkin, "From cohomology in physics to q-connectivity in social science," *International Journal of Man-Machine Studies*, vol. 4, no. 2, pp. 139–167, 1972.
[21] H. Riihimäki, "Simplicial $\boldsymbol{q}$-connectivity of directed graphs with applications to network analysis," *SIAM Journal on Mathematics of Data Science*, vol. 5, no. 3, pp. 800–828, 2023.
[22] SciPy Clustering package, "Perform hierarchical/agglomerative clustering," https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.linkage.html, [Online; accessed 22-February-2024].
[23] D. Sculley, "Web-scale k-means clustering," in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW '10. Association for Computing Machinery, 2010, p. 1177–1178.
[24] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, pp. 395–416, 2007.
[25] Scikit-Learn Library, "Gaussian mixture models," https://scikit-learn.org/stable/modules/mixture.html#gmm, [Online; accessed 14-February-2024].