Benefits of hypergraphs for density-based clustering
Louis Hauseux, Konstantin Avrachenkov, Josiane Zerubia

To cite this version:
Louis Hauseux, Konstantin Avrachenkov, Josiane Zerubia. Benefits of hypergraphs for density-based clustering. EUSIPCO 2024 - 32nd IEEE European Signal Processing Conference, Aug 2024, Lyon, France. hal-04617936

HAL Id: hal-04617936
https://hal.science/hal-04617936
Submitted on 19 Jun 2024

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.

Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License
Benefits of hypergraphs for density-based clustering

Louis HUASEUX  
*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].](image)

Precise discretization of the Euclidean 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 $f$ 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 II-A). We show that the $K$-Polyhedra match with the high-density clusters of the density estimator $f_{K-NN}$ (see Theorem in Section III-C).

II. CLASSICAL MATHEMATICAL MODEL

Assume that the dataset $\mathcal{X}_n := \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ 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 $L_r$.

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

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 & SOCOCOLA [12] for a much more extensive presentation of hierarchical clustering.
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 $X \subset E$ in the Euclidean space $\mathbb{R}^d$, the geometric graph $G(X, r)$ of radius $r$ is the graph whose nodes are the points $x \in X$ and there is an edge between $x, y \in X$ if $||x - y|| \leq r$, where $||\cdot||$ denotes the Euclidean distance (see Penrose [4]).

B. Single-Linkage $\simeq$ Geometric graphs $\simeq f_{1}$-Nearest Neighbor

Single-Linkage algorithm constructs a hierarchical clustering as follows: It starts with the trivial initial clustering (n points for n clusters) $C_0 = \{C_0^1, \ldots , C_n^0\}$ with $C_i^0 = \{x_i\}$. At each step, we merge the two clusters that are the closest for the distance: $d_{\text{Clust}}(C, C') = \min_{x \in C, y \in C'} ||x - y||$.

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

Furthermore, we propose a new clustering algorithm, calling ‘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 $G(X_{300}, 0.07)$ has a giant component almost corresponding to $A \cap X_{300}$.

Hartigan [13] showed that the Single-Linkage algorithm is a consistent estimator of high-density clusters in dimension $d = 1$, but only fractionally 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 $x \in X_n$ having more than $K$ neighbors in their $r$-neighbourhood are considered:

$$X_n^{K,r} := \{ x \in X_n : |B(x, r) \cap X_n| \geq K \} \subset X_n.$$  

Then a geometric graph $G(X_n^{K,r}, r)$ is constructed on $X_n^{K,r}$ 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 $x, y \in X_n^{K,r}$ 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 $f_{K,NN}$ are quite different from those of the Robust Single-Linkage (or (H)DBSCAN).

Look at the Fig. 4 for such an example. The discrete high-density clusters at level $r$ of the point cloud $X_n$ 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 $H_{f_{NN}}$ 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, $\{A, B, C, D, E, F\}$.

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...
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 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 \( X \) and a set of edges \( E \). The difference is that the edges \( e \in E \) are not only pairs \( \{x, y\} \) of vertices \( x, y \in X \) but can be any non-empty subsets of the vertices \( e \subseteq X \). 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 \( e = \{x_{i0}, \ldots, x_{ik}\} \) 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 \( K \).

2The 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]).

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 \( \hat{C}(X_n, r) \) is the hypergraph whose \( K \)-dimensional hyperedges \( e = \{x_{i0}, \ldots, x_{ik}\} \) are in \( \hat{C}(X_n, r) \) if their epicentrum \( E(e) \) is not empty:

\[
E(e) := \bigcap_{j=0}^{k} \overline{B}(x_{ij}, r/2) \neq \emptyset.
\]

Note that the 1-skeleton of \( \hat{C}(X_n, r) \) corresponds to the geometric graph \( G(X_n, r) \).

C. Correspondence between K-Polyhedra of \( \hat{C}(X_n, r) \) and the high-density clusters of \( f_{KNN} \)

Let \( X_n \subset \mathbb{R}^d \) be a point cloud IID generated by a continuous density function \( f \). Let \( L^K_r \) be the \( r \)-level set of the \( K \)-Nearest Neighbors density estimator

\[
L^K_r = \{ x \in \mathbb{R}^d : |\overline{B}(x, r/2) \cap X_n| \geq K \} = \bigcup_{e \in \hat{C}(X_n, r)} E(e)
\]

and \( H_{f_{KNN}}(r) \) the associated high-density clusters, i.e. the set of connected components of \( L^K_r \).

Note that for fixed radius \( r \), \( L^K_{r+1} \subset L^K_r \) we can therefore ‘link’ the \( (K+1) \)-clusters \( C \in H_{f_{KNN}}(r) \) which are in the same \( K \)-cluster \( C' \in H_{f_{KNN}}(r) \). Let us define

\[
\text{Link}^K_r \left( C' \in H_{f_{KNN}}(r) \right) = \left\{ C \subset C' : \bigcup_{C \in C'} E(e) \right\} \setminus \{\emptyset\}.
\]

We define the \( \epsilon \)-dilation of the set \( X \subset \mathbb{R}^d \) by

\[
\delta_{\epsilon}(X) = \{ x \in \mathbb{R}^d : \overline{B}(x, \epsilon) \cap X \neq \emptyset \}.
\]

**THEOREM.** Let \( K \) be the dimension of the polyhedra, \( r \) a radius, \( \hat{C}(X_n, r) \) the Čech complex of radius \( r \) on the cloud \( X_n \) and \( V_1, \ldots, V_n \) the sets of vertices of its \( \alpha \) K-Polyhedra. Then, \( \alpha = |\text{Link}^K_r| \). Moreover, the \( V_i \) are exactly given by

\[
\delta_{r/2}(C) \cap X_n \text{ for } C \in \text{Link}^K_r.
\]

By looking at the K-Polyhedra, we can therefore identify the clusters of \( H_{f_{KNN}}(r) \) that are connected in \( L^K_r \). While Robust Single-Linkage (or HDBSCAN) merges all the clusters.
of $H_{f_{K+1,s_0}}(r)$ 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 $\check{C}(X_n, r)$ 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 $\check{C}(X, r)$. 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:** $X$ the point cloud, $K \in \mathbb{N}$ and $\text{PercolThreshold} \in \mathbb{N}$

**Output:** Hierarchical clustering $\hat{H}: r \mapsto \hat{H}(r)$

for $r$ from 0 to $+\infty$ do

$\text{Hypergraph} \leftarrow \check{C}(X, r)$

$\text{Polyhedra} \leftarrow K$-Polyhedra($\text{Hypergraph}$)

for $\text{Polyhedron}$ in $\text{Polyhedra}$ do

if Length($\text{Polyhedron}$) $< \text{PercolThreshold}$ then

Remove($\text{Polyhedron}$) from $\text{Polyhedra}$

end if

end for

$\hat{H}(r) \leftarrow \text{Polyhedra}$

end for

Return $\hat{H}$

Once a hierarchical clustering $\hat{H}(r)$ 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]

$X_{500}$ is composed of 3 blobs (see Fig. 5 and 7). The computation of the $K$-Polyhedra on the Čech complex $\check{C}(X_{500}, r)$ is made with the same parameters as in HDBSCAN: $K \leftarrow 2$ and the percolation threshold $\text{PercolThreshold} \leftarrow 15$. 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 $X_{500}$ 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 unsupervised – 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).

#### 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 $\text{PercolThreshold} \leftarrow 30$ 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 (4th in Tab. 11) 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. 11 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]).

Note that here, a ‘clustering’ $H(r)$ may not be a partition of $X$: some points at the frontier of $K$-Polyhedra may appear in several clusters.
V. CONCLUSION AND PERSPECTIVES

This paper presents a new clustering method called ‘Hypergraph-Perczel’ for data X ⊂ Rd in the Euclidean space.

We construct hypergraphs (Čech complexes) on X and use a restrained notion of connectivity called the K-Polyhedron connectivity (Definition III-A). We show that our ‘Hypergraph-Perczel’ 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