



HAL
open science

Hypergraphs, percolation, and hierarchical clustering

Louis Hauseux, Konstantin Avrachenkov, Josiane Zerubia

► **To cite this version:**

Louis Hauseux, Konstantin Avrachenkov, Josiane Zerubia. Hypergraphs, percolation, and hierarchical clustering. Complex Networks 2024 - 13th International Conference on Complex Networks and their Applications, Dec 2024, Istanbul, Turkey. hal-04723052

HAL Id: hal-04723052

<https://inria.hal.science/hal-04723052v1>

Submitted on 9 Oct 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

Hypergraphs, percolation and hierarchical clustering

Louis Hauseux ^{**}, Konstantin Avrachenkov , and Josiane Zerubia 

Inria, Université Côte d’Azur, Sophia-Antipolis, France.
FirstName.LastName@inria.fr,

Abstract. We are interested in, firstly, measuring the theoretical performance of hierarchical clustering algorithms which depend on a scale parameter; and secondly, in improving the State-of-the-Art of this family of hierarchical clustering with respect to our new measure.

Single-Linkage is perhaps the simplest and the most famous algorithm belonging to this family. Nowadays, the State-of-the-Art clustering algorithm, HDBSCAN, works in a similar way to Single-Linkage (with some refinements we will study later).

Herein, we explain why the percolation phenomenon is omnipresent behind this family of clustering algorithms. Previously, we defined an index which we have named *percolation rate* to measure the theoretical capability of algorithms to identify different high-density levels.

In this paper, we show that using hypergraphs is a natural way to generalize Single-Linkage with higher-order interactions (not just ‘single’). New high-order connected components on hypergraphs, we call *K-polyhedra*, have much better percolation rates than the classic ‘robustification’ of Single-Linkage (used *e.g.* by HDBSCAN), the *K-Robust Single-Linkage* components. We investigate in detail the important cases of \mathbb{R}^2 and \mathbb{R}^3 for $K \in \{1, 2, 3\}$.

Keywords: hierarchical clustering, percolation, hypergraphs, performance analysis

1 Introduction

Hierarchical clustering [30] is a cluster analysis technique aimed at constructing a hierarchy of clusters we can represent by a tree (see FIG. 1).

In this paper, we suppose that the level of the hierarchy in the tree depends on a parameter (*e.g.* a density ρ , or a distance/radius r). Varying this parameter, we can observe the structures (= clusters) appearing and then disappearing (merging). We call this kind of process “persistent analysis”, by analogy with the *persistent homology* in algebraic topology [2].

^{**} 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-17-EURE-0004) and 3IA Côte d’Azur for partial funding of his PhD thesis.

For example, Single-Linkage algorithm (presented in Section 1.1) performs hierarchical clustering by grouping iteratively the two closest clusters. Single-Linkage algorithm has received numerous improvements. The State-of-the-Art in clustering today is held by HDBSCAN (Section 2.2), which is essentially based on the Robust Single-Linkage algorithm (Section 2.1). We devote Section 2 to the study of these proposed improvements which led to HDBSCAN.

It is worth noting that (Robust) Single-Linkage can be seen from persistent analysis: Each clustering level of the hierarchy corresponds exactly to the connected components of a geometric graph [28]. Adopting this point of view allows us to observe the mathematical phenomenon behind this family of algorithms. As the level increases, small structures appear and then merge to form giant structures: this is the phenomenon of *percolation* (see Section 3.1).

The speed of percolation depends on the type of objects we are considering (graphs, hypergraphs, etc.) and the definition we take for connected components (usual definition, K -polyhedra defined in DEF. 3, etc.). The faster this speed, the better the algorithm will be able to distinguish between neighboring high-density clusters (DEF. 1, Section 1.2). These considerations lead us to define the *percolation rate* (DEF. 2, Section 3.3) in the previous Proc. of Complex Networks [18].

In this paper, we go much further. First, we explain why the mathematical phenomenon of percolation is directly linked to the performance of this family of algorithms (Section 3.2). In another short paper [16], we calculated this percolation rate on a discretized K -Nearest Neighbors density estimator. We obtained a kind of consistency when $K \rightarrow \infty$.

Herein, we are interested in one of the perspectives mentioned in last year's Complex Networks paper [18]: Look at what happens in terms of percolation rate if we change the notion of connected component. To do this, we generalized in a previous paper [17] the notion of connected component on hypergraphs – more precisely: on simplicial complexes. We called K -polyhedra (see DEF. 3) this new notion of *more constrained* connected components. We already showed in [17] that K -polyhedra were the right generalization to consider for Single-Linkage: Unlike the Robust Single Linkage components, the K -polyhedra correspond to the high-density clusters of the K -Nearest Neighbors density estimator. At the end of the present paper (Section 3.3), we compute the percolation rate of K -Čech polyhedra and K -Robust Single-Linkage components ($K \in \{1, 2, 3\}$, $K = 1$ being the case of geometric graphs) in \mathbb{R}^d , with $d \in \{2, 3\}$. We analyze the results and show new theoretical advantages of K -polyhedra.

1.1 Single-Linkage: Definition

Single-Linkage [10, 11] is a very simple, efficient and widely used hierarchical clustering algorithm [30].

Let \mathcal{X}_n be a set of n data in any metric space, the Single-Linkage algorithm constructs a hierarchical tree (SLHT), called *dendrogram* (see FIG. 1), as follows: It starts with the trivial initial clustering (n points for n clusters) $C^0 = \{C_1^0, \dots, C_n^0\}$ with

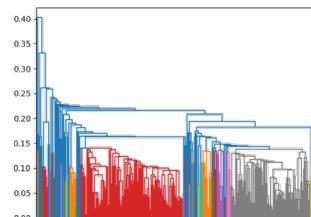


Fig. 1. \mathcal{X}_{350} (FIG. 3) SLHT. Colours denote the clustering at level $2r \leftarrow 0.14$.

$C_i^0 = \{x_i\}$. At each step, it merges the two clusters that are the closest for the distance:

$$d_{\text{Clust}}(C, C') = \min_{x \in C, x' \in C'} \|x - x'\|.$$

At step t , the resulting clustering $\mathcal{C}^t = \{C_1^t, \dots, C_{n-t}^t\}$ is made of $n - t$ clusters. So, at step $t = n - 1$, the final clustering $\mathcal{C}^{n-1} = \{\mathcal{X}_n\}$ is made of only one cluster encompassing all the points.

Observe that Single-Linkage clustering can be recovered using the connected components of a geometric graph [28] built on the data¹. The geometric graph $\mathcal{G}(\mathcal{X}_n, r)$ of radius r is the graph whose nodes are the points $x \in \mathcal{X}_n$ and there is an edge between $x, x' \in \mathcal{X}_n$ if

$$\|x - x'\| \leq 2r.$$

This insight will be especially useful later in the article, as it links Single-Linkage to the *percolation* (see Section 3); allowing us to theoretically measure its performance and compare it to other percolation algorithms.

1.2 Mathematical model

Assume that the dataset $\mathcal{X}_n := \{x_1, \dots, x_n\} \subset \mathbb{R}^d$ is a cloud of n points all independently and identically distributed (i.i.d.) according to a probability measure with density $f : \mathbb{R}^d \rightarrow \mathbb{R}_+$.

Density-based clustering. Considering the point generation density f is a natural way of tackling our problem [5, 30]. With the very intuitive idea that the different clusters are represented by the ‘peaks’ [32] of the density function f , HARTIGAN [13] defined the *high-density clusters* of f at level ρ . By varying the level ρ , we can obtain a hierarchical clustering (cf. FIG. 2).

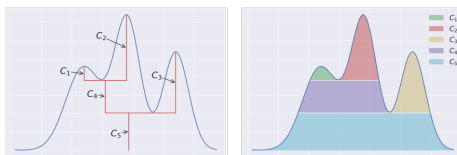


Fig. 2. Hierarchical clustering: clusters and their *relative excess of mass* $E_R(C)$ [6, 22] (coloured areas). © Images taken from [22].

Definition 1 (High-density clusters). *The high-density clusters $H_f(\rho)$ of the density function f at level ρ are the connected components of the level set L_ρ*

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

2 State-of-the-Art: Improvements of Single-Linkage

Many improvements of Single-Linkage have been proposed. We are particularly interested in the HDBSCAN algorithm, one of today’s best clustering algorithms. HDBSCAN is itself an improvement of K -Robust Single-Linkage (Section 2.1).

¹ Although this result seems to be known, since it is used by *e.g.* by [28], to the best of our knowledge it does not seem to have been fully demonstrated. The full proof will be found in a journal paper we are currently writing.

2.1 Robust Single-Linkage

The clusterings of Single-Linkage match exactly with the high-density clusters of the 1-Nearest Neighbor density estimator [17]. However, the 1-NN estimator can be highly irregular. In order to make the results more ‘robust’ (*i.e.* smoother as well as having good consistency properties when $n \rightarrow +\infty$), WISHART proposed a *robust* version of the Single-Linkage [7, 35], inspired by the consistency of the K -Nearest Neighbors density estimator [1]. The difference is: In the K -Robust version, a point $x \in \mathcal{X}_n$ must have at least K points in its r -neighbourhood to appear in the geometric graph.

2.2 HDBSCAN

HDBSCAN [6, 22] brings two major improvements to Robust Single-Linkage.

The relative excess of mass criterion. HDBSCAN introduces a criterion for multi-scale clustering based on the *excess of mass* of a cluster (see [15, 27]). *Cf.* FIG. 2 for an illustration: the *relative excess of mass* $E_R(C)$ of a high-density cluster C is the area of the coloured zones. The final clustering $\mathcal{C} = \{C_1, \dots, C_\alpha\}$ returned by HDBSCAN is the one maximizing the sum of $\sum_{i=1}^\alpha E_R(C_i)$.

These quantities can be easily estimated. In fact, suppose we have an estimator $\hat{\lambda}_x$ of $f(x)$. Let $C \in H_f(\lambda)$, $E(C)$ can be estimated by $\hat{E}(C) \propto \sum_{x \in C} (\hat{\lambda}_x - \hat{\lambda})$.

Note that, in HDBSCAN, $\hat{\lambda}_x = \frac{1}{r_x}$ where r_x is the smallest radius for which $x \in \mathcal{X}_n$ is clustered. This estimator can be applied in any metric space. However, when the point cloud $\mathcal{X}_n \subset \mathbb{R}^d$ takes its values in the Euclidean space \mathbb{R}^d of dimension d , it is more relevant to estimate λ_x by

$$\hat{\lambda}_x = \left(\frac{1}{r_x} \right)^d.$$

The percolation threshold. HDBSCAN prunes the cluster tree by only keeping clusters that have a size greater than a new parameter. In doing so – and without saying it explicitly –, the authors of HDBSCAN introduced a kind of *percolation threshold*: only large components are taken into account, eliminating the ‘noise’ of small components appearing randomly.

3 Percolation phenomenon

The great interest of observing clustering algorithms in terms of persistent analysis is that we can observe the mathematical phenomenon behind: the *percolation*.

This section begins with a brief theoretical introduction to percolation (Section 3.1). We then show (Section 3.2) that percolation is at the heart of the Single-Linkage algorithm (and our K -generalization [17]).

But all is not lost: If we cannot hope to *perfectly* recover high-density clusters, we can hope to recover a *fraction* of them. That is what the index we defined, the *percolation rate*, measures (DEF. 2, Section 3.3).

Finally, in Section 3.3, in low-dimensional spaces (\mathbb{R}^2 and \mathbb{R}^3), we empirically compute this percolation rate and compare K -polyhedra *vs.* K -Robust Single-Linkage components.

3.1 Introduction to percolation

The latin verb ‘percolare’ means *to strain through, to filter*. Water *percolates* through a rock if it can infiltrate and pass through small pores.

The mathematical percolation model was first introduced by BROADBENT & HAMMERSLEY [4] in 1957 to answer the question: can water seep through a rock? See the survey of DUMINIL-COPIN [8] for a much more detailed introduction.

Percolation is a phenomenon which, under local constraints, can be observed macroscopically. It is the precise moment when macro-structures appear. This study is very interesting for modelling and studying numerous problems from everyday life: the spread of a forest fire, the existence of a giant component in a wireless network, blood coagulation (platelet percolation), etc.

When the space is continuous – like the Euclidean space \mathbb{R}^d –, we speak of *continuum percolation* [24, 28]. Geometric graphs with point cloud $\mathcal{X}_n \subset \mathbb{R}^d$ in the Euclidean space are a particular case of continuum percolation.

Let us present a definition of percolation inspired by PENROSE [28] and a bit modified to be compatible with higher-order graphical interactions.

Let $\mathcal{X} := \mathcal{H}^\lambda$ be a Poisson point process of intensity λ on the torus $\mathbb{T}^l := \mathbb{R}^d / l\mathbb{Z}^d$ of size l and $\mathcal{X}^0 := \mathcal{X} \cup \{0\}$. Let $\mathcal{C} = \{C_1, \dots, C_o\}$ be a clustering on \mathcal{X} (not necessarily a partition, some points may be unclustered; other points may be multi-clustered). In continuum percolation, \mathcal{C} is often the clustering given by the connected components of the geometric graph $\mathcal{G}(\mathcal{X}, \frac{1}{2})$. For our purpose, we consider the K -polyhedra of Čech complexes $\check{C}(\mathcal{X}, \frac{1}{2})$ (*cf.* [17]). Denote $M_l^\lambda := \max_i |C_i|$ the (random variable) size of the largest cluster. In the same way, denote $M_l^{\lambda,0} \geq M_l^\lambda$ the size of the largest component on $\mathcal{X}^0 \supset \mathcal{X}$. Define the probability for a new point to be clustered in the largest cluster as $p_l(\lambda) := \mathbb{P}_{l,\lambda} \left[M_l^{\lambda,0} > M_l^\lambda \right]$.

Finally, we define the *percolation probability* with intensity λ and associated it to the clustering method we look at:

$$p_\infty(\lambda) := \liminf_{l \rightarrow +\infty} p_l(\lambda).$$

The probability $p_\infty(\lambda)$ can be seen as the proportion of points lying in the largest component. It is a non-decreasing function in λ . (For all the properties on the *probability percolation* function $p_\infty(\cdot)$, look at the monograph of MEESTER & ROY [24]. The properties apply also to K -Čech polyhedra [17].) Moreover, there exists a critical value λ_c for which:

$$p_\infty(\lambda) = 0 \text{ if } \lambda < \lambda_c \quad \text{and} \quad p_\infty(\lambda) > 0 \text{ if } \lambda > \lambda_c.$$

The critical value λ_c depends on the objects, the kind of connectivity and also the dimension d of the space. This is the threshold for the appearance of a

giant component. Let us illustrate this idea with an example (see FIG. 3) on a $n = 350$ point cloud \mathcal{X}_{350} . There are two high-density clusters, the squares $A = [-1.5, -0.5] \times [-0.5, 0.5]$ and $B = [0.5, 1.5] \times [-0.5, 0.5]$. The density f is constant on each of the three areas: the two squares A and B , and the background. By varying the radius r of the geometric graph $\mathcal{G}(\mathcal{X}_{350}, r)$, we observe phase transitions around certain critical radii r_c (depending on the density of the area). For $r < r_c$, there are only small connected components. For $r > r_c$, there is one (unique) giant component encompassing virtually all the points of the cluster.

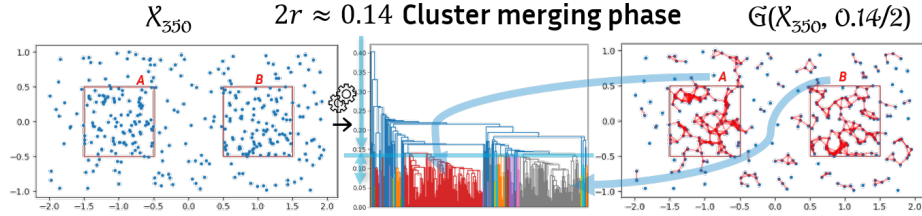


Fig. 3. From left to right: 1) The point cloud \mathcal{X}_{350} . 2) The dendrogram of the Single-Linkage applied on \mathcal{X}_{350} . 3) The geometric graph $\mathcal{G}(\mathcal{X}_{350}, r \leftarrow 0.14/2)$.

3.2 Fractional Consistency of the Robust Single-Linkage

HARTIGAN [14] showed that Single-Linkage is a consistent estimator of high-density clusters in dimension $d = 1$, but only *fractionally* consistent in dimension $d \geq 2$. The reason behind this is that dimension $d = 1$ is very specific: it is the only dimension where the phenomenon of continuum percolation [24, 28] does not occur. In \mathbb{R}^d with $d \geq 2$, as the radius increases, giant connected components will appear almost surely. The problem is that these giant components will merge before covering all the points in their cluster. We can only hope to recover a *fraction* of the points.

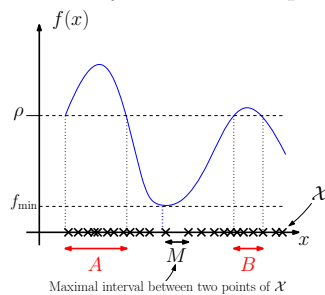


Fig. 4. Largest empty space M between two points.

Single-Linkage is consistent on \mathbb{R}^d for $d = 1$.

Let $A, B \in H_f(\rho)$ be two disjoint high-density clusters of f at level ρ . With probability $p_n \rightarrow 1$ when $n \rightarrow +\infty$, there exists a level r_n in the dendrogram with two disjoint clusters $C_A^{r_n}$ and $C_B^{r_n}$ such that each of the two clusters contains respectively all the points in A and in B :

$$C_A^{r_n} \supseteq A \cap \mathcal{X} \quad \text{and} \quad C_B^{r_n} \supseteq B \cap \mathcal{X}.$$

The argument is simple: the location of the largest empty space M between two consecutive points x_i and x_j on the real axis will converge (in probability) to $\operatorname{argmin} f$ [12]. See FIG. 4 for an illustration. It follows that there is a ‘cut-off’ at this location for $r \lesssim M$: We obtain a cluster containing the points of A and another cluster containing those of B .

For higher dimensions, Single-Linkage is not consistent. Nevertheless, it can be *fractionally consistent*.

Single-Linkage is not consistent on \mathbb{R}^d for $d \geq 2$. Let us see the inconsistency of Single-Linkage on a discretized model. Let $\Lambda := \mathbb{Z}^2$ be the grid. We perform on Λ site percolation [3].

A and B (the two high-density clusters) are two $n \times n$ squares aligned and n apart. The cloud is a Poisson point process with intensity λ outside the two squares and $\lambda' > \lambda$ on the two squares. As soon as a point of the cloud \mathcal{X} falls into one of the cells, it becomes activated (*open sites* are drawn in red in FIG. 5 and 6). We are therefore in the case of an (independent and inhomogeneous) site percolation model [3].

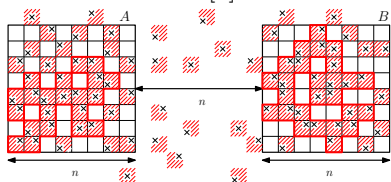


Fig. 5. Percolation occurs on A and B .

As λ and λ' increase (in a proportional way), the cloud increases too. At one point², percolation occurs on A and B : Giant components encompassing a positive fraction $\theta(p')$ – the discrete equivalent of $p_\infty(\cdot)$ – of the sites in A and B is present almost surely; we can recover the sites of these two giant clusters. See FIG. 5: the two giant clusters in A and B are surrounded in red.

The problem is when we want to recover *all* the sites of the clusters A and B . If the cloud continues to grow, before recovering entirely A and B , percolation will occur on the whole lattice \mathbb{Z}^2 . Thus, a ‘brige’ appears between A and B and the two clusters merge (*cf.* FIG. 6).

Our goal is to find the kind of objects which connected components have the largest possible recoverable ‘fraction’.

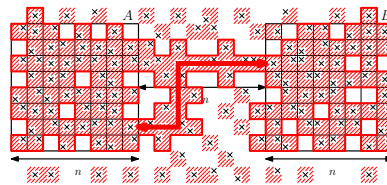


Fig. 6. Percolation occurs everywhere \implies Clusters within A and B merge.

3.3 The percolation rate

How can we measure this recoverable ‘fraction’? This led us to define a *percolation rate* in our previous article [18]. We are now going to take it a step further: Section 3.3 is the main contribution of the current paper.

Percolation is a ‘fast’ phenomenon. Let \mathcal{H}^λ be a Poisson point process on \mathbb{R}^2 with intensity λ and consider the geometric graph $\mathcal{G}(\mathcal{H}^\lambda, \frac{1}{2})$ of radius $2r = 1$. For $\lambda < \lambda_c$, there is (almost surely) no infinite component. Then, the proportion of points lying in the largest component is null: $p_\infty(\lambda) = 0$. For $\lambda > \lambda_c \approx 1.44$ [29, 36], percolation occurs. A giant component appears. As soon as the giant component appears, if we increase the intensity λ slightly, the probability of percolation $p_\infty(\lambda)$ approaches 1 quickly. At this point, the giant component includes almost all the points ($p_\infty(\lambda) \lesssim 1$).

How can we measure the speed of percolation? Let us take the (continuous) example of FIG. 3. There are two high-density clusters A and B , two unit cubes

² When $p' > p_c \approx 0.592746$ [25].

in \mathbb{R}^d . The point cloud \mathcal{X} is composed of a Poisson Point Process of intensity $n\lambda_1$ on $A \cup B$ and $n\lambda_2$ on $\mathbb{R}^2 \setminus (A \cup B)$, with $\lambda_1 > \lambda_2$. Asymptotically, when $n \rightarrow \infty$, the two clusters C_A and C_B of the geometric graph $\mathcal{G}(\mathcal{X}, r_n)$ will merge when we take a radius r_n such that percolation takes place on \mathbb{R}^2 as a whole, that is when $n(2r_n)^d \lambda_2 = \lambda_c$. At this precise moment, the intensity (relative to the radius r_n) on $A \cup B$ is given by $n(2r_n)^d \lambda_1 = \lambda_c \frac{\lambda_1}{\lambda_2}$. This means that the *fraction of points from A (resp. B) recovered by C_A (resp. C_B)* is:

$$1 - \varepsilon := p_\infty \left(\lambda_c \frac{\lambda_1}{\lambda_2} \right) = \lim_{n \rightarrow \infty} \frac{|C_A \cap A|}{|\mathcal{X} \cap A|}.$$

If we now fix a *sensitivity* (or *recall*) we want to get, e.g. $1 - \varepsilon = 95\%$. Let $\lambda_{\max} := p_\infty^{-1}(1 - \varepsilon)$ the intensity for which this sensitivity is obtained. This sensitivity is theoretically achievable if and only if $\frac{\lambda_c}{\lambda_{\max}} > \frac{\lambda_2}{\lambda_1}$. This could lead us to take $\frac{\lambda_c}{\lambda_{\max}}$ as percolation rate. The larger this quantity, the more the clusters C_A and C_B will recover an important fraction of A and B before merging.

However, for practical reasons, this quantity is difficult to estimate because we do not have the function $p_\infty(\cdot)$ but only an approximation. For the types of clustering we are going to look at, what is its critical value λ_c ? or similarly: When does the curve of $p_\infty(\cdot)$ pass above 0? This question is not so simple (see, for example, FIG. 9). It is simpler to approximate this quantity by $\lambda_{\min} := p_\infty^{-1}(\varepsilon)$, the intensity for which we can ‘detect’ a cluster.

Definition 2 (Percolation rate). Let $\mathcal{C}(\mathcal{X}, r \leftarrow 1/2)$ be a hierarchical clustering method with the parameter r fixed to $\frac{1}{2}$. Let $\mathcal{X}_\lambda := \mathcal{H}_\lambda$ be a Poisson point process on \mathbb{R}^d of intensity λ and $p_\infty(\lambda)$ be the percolation probability function associated to the clustering method $\mathcal{C}(\mathcal{X}_\lambda, \frac{1}{2})$ (see Section 3.1). Let $\varepsilon \ll 1$ be the level of sensitivity expected. Let $\lambda_{\min} := p_\infty^{-1}(\varepsilon)$ and $\lambda_{\max} := p_\infty^{-1}(1 - \varepsilon)$ be respectively the thresholds of detection and recovering of the giant clusters. The quantity of interest – i.e. our percolation rate v – is defined by

$$v := \frac{\lambda_{\min}}{\lambda_{\max}} \in (0, 1).$$

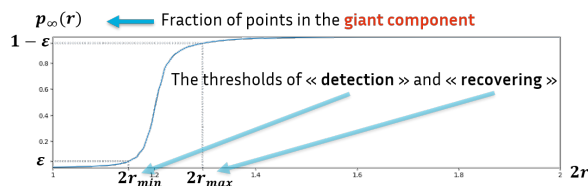


Fig. 7. Percolation probability $p_\infty(r)$ in \mathbb{R}^2 by simulations of giant random geometric graphs. © [33].

we changed the variable $2r \leftarrow \sqrt{\lambda}$ from intensity λ to radius r (previously fixed to $2r = 1$) for a more intuitive vision of percolation. Note that $p_\infty(r)$ is positive

The faster the percolation is, the closer is the ratio $v := \frac{\lambda_{\min}}{\lambda_{\max}}$ to 1. Note that we could have defined the *percolation rate* with radii r and not intensity λ . In \mathbb{R}^d , the two notions being linked by the formula $v_{\text{radii}} := \frac{r_{\min}}{r_{\max}} = \left(\frac{\lambda_{\min}}{\lambda_{\max}} \right)^{\frac{1}{d}} = v^{\frac{1}{d}}$.

On the curve of FIG. 7,

even if $2r \lesssim 2r_c = \sqrt{\lambda_c} \approx 1.2$; this is due to the approximation of \mathbb{R}^2 by a finite square 251×251 . Moreover, the real curve starts with a positive slope [9, 20, 21]:

$$\exists C > 0, \forall r \geq r_c, \quad p_\infty(r) \geq C(r - r_c).$$

Percolation rate for the K -polyhedra.

K-polyhedra on hypergraphs Robust Single-Linkage has theoretical weaknesses we showed in another article [17]: It introduces strong constraints on vertices but has still relaxed constraints on edges (still ‘single-link’). To tackle this issue, we propose to use *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 new kind of connected components we call K -polyhedra (see FIG. 8).

We show in this section that the natural way to generalize geometric graphs is to use Čech complexes with a more constrained notion of connected component we call a K -polyhedron (DEF. 3).

Simplicial complexes [26] are a sub-family of hypergraphs where the presence of a hyperedge implies the presence of its sub-hyperedges. For example, the presence of a tetrahedron (hyperedge of 4 vertices) implies the presence of its 4 triangles and of its 6 edges.

Čech complexes are the natural way to generalize geometric graphs. Given a point cloud $\mathcal{X} \subset \mathbb{R}^d$ in the Euclidean space, a $(K + 1)$ -hyperedge $\sigma = \{x_{i_0}, \dots, x_{i_K}\}$ is in the Čech complex $\check{C}(\mathcal{X}, r)$ of radius r if there exists a ball of radius r encompassing all the $K + 1$ points.

Note that the restriction of $\check{C}(\mathcal{X}, r)$ to the edges (2-hyperedges) gives the geometric graph $\mathcal{G}(\mathcal{X}, r)$.

Definition 3 (K -polyhedra). A K -polyhedron is defined inductively:

- The convex hull of a $(K + 1)$ -hyperedge $\sigma = \{x_{i_0}, \dots, x_{i_K}\}$ is a K -polyhedron.
- If two K -polyhedra share a common facet (hyperedge of size K), then their union is still a polyhedron of dimension K .

See FIG. 8 for an illustration of the 2-polyhedra (“Triangle-connected” components).

We showed [17] that K -polyhedra fit much better with the K -high-density clusters than K -Robust Single-Linkage components do. Now we illustrate yet in another way that K -polyhedra clustering has better performance than K -Robust Single-Linkage: in terms of *percolation rate* v .

In order to compute the percolation rate of a clustering algorithm, we need first to compute its percolation probability $p_\infty(\lambda)$.

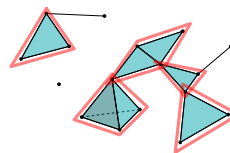


Fig. 8. 2-polyhedra on a hypergraph. 2-connected components are highlighted with red colour.

Methodology for computing the percolation probability $p_\infty(\cdot)$. In order to compute easily the K -polyhedra of Čech complexes $\check{C}(\mathcal{X}, r \leftarrow \frac{1}{2})$ (see [17]) on a huge point cloud \mathcal{X} , we work in Euclidean spaces \mathbb{R}^d of small dimensions: $d \in \{2, 3\}$. It allows us to discretize the space (the unit is cut in 2δ intervals), having a *Density* map containing the number $Density(x) := |B(x, \frac{1}{2}) \cap \mathcal{X}|$ of r -Neighbors. We then use the THEOREM of [17] to obtain the K -polyhedra. Finally, we compute the *percolation probability* $p_i(n)$ on the torus $\mathbb{R}^d/l\mathbb{Z}^d$ according to the definition we gave in Section 3.1 by generating progressively m different point clouds \mathcal{X}_n with $n \in \{1, \dots, N\}$, $N := \lambda_{\text{sup}} \times l^d$.

At this moment, we use the trick explained by BOLLOBÁS & RIORDON pp. 175-177 [3]. To obtain statistics $p(\mu)$ for an intensity $\mu \in I$, we compute first the statistics $p(n)$ for a fixed size $n \in \{1, \dots, N\}$ of the point cloud $\mathcal{X}_n \subset \mathbb{T}^l := \mathbb{R}^d/l\mathbb{Z}^d$, with $N \gg \mu$. At the end, if $|\mathcal{X}_n| \sim \text{Poisson}(\mu)$, then we can compute easily $p(\mu) = \sum_{n=1}^{\infty} p(n) \times \mathbb{P}[|\mathcal{X}_n| = n]$.

Two big advantages of this method. First, the statistics $p(n)$ computed on a point cloud \mathcal{X}_n are useful to compute $p(\mu)$ for all $\mu \in I$. Second, once the statistics on \mathcal{X}_{n-1} have been computed, those on \mathcal{X}_n can be efficiently computed.

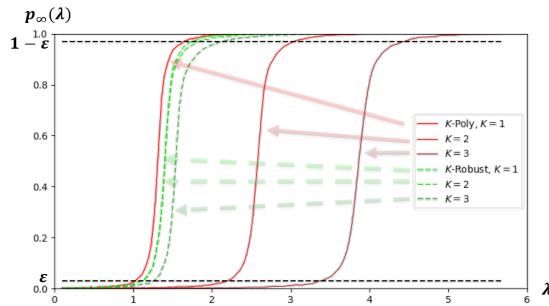


Fig. 9. Percolation probability for K -polyhedra (red-plain) and K -Robust Single-Linkage (green-dashed) in $\mathbb{T}^l = \mathbb{R}^2/l\mathbb{Z}^2$ (with $l = 100$).

Results in \mathbb{R}^2 . For $K \in \{1, 2, 3\}$, we computed the percolation probability of the K -polyhedra with discretization of the space (see previous Section 3.3) and the exact K -Robust Single-Linkage components.

We took the parameters $l = 100$, $\delta = 8$, $\lambda_{\text{sup}} = 6$, $m = 24$ and obtained the curves of FIG. 9. The K -polyhedra are represented in red and the K -Robust Single-Linkage components

in green. The ‘1-polyhedra’ is almost identical to the ‘1-Robust components’ curve. They are both theoretically equal to that of the geometric graph components (compare with FIG. 7); the difference coming from the discretization of the space.

Note that 2-Robust Single-Linkage components correspond to those of the geometric graph, pruned of the vertices with only one neighbor, that is pruned of the leaves. It is also very close to the 1-Robust Single-Linkage components.

The 3-Robust Single-Linkage components are quite different and percolate for larger intensity: $\lambda_c^{3\text{-Robust}} > \lambda_c^{2\text{-Robust}} = \lambda_c^{1\text{-Robust}}$. But not as large as the intensity $\lambda_c^{3\text{-poly}} \gg \lambda_c^{2\text{-poly}} \gg \lambda_c^{3\text{-Robust}}$ needed for the 2- or 3-polyhedra. The latter notion of connected components is much more restrictive and percolation occurs later. With these percolation probability functions, we are now able to compute and compare the percolation rate v . Results are given in Tab. 1. The first row ($K = 1$) shows similar results ($v^{1\text{-Robust}} = 0.64$ vs. $v^{1\text{-poly}} = 0.62$). This

Table 1. Percolation rate v^K in \mathbb{R}^2 for K -polyhedra and K -Robust Single-Linkage components. $K \in \{1, 2, 3\}$, $v^K := \frac{\lambda_{\min}^K}{\lambda_{\max}^K}$ with $\varepsilon = 3\%$. Cf. Fig. 9.

Clustering	K -Robust Single-Linkage	K -polyhedra
$K = 1$	1.12/1.75 = 0.64	1.04/1.67 \approx 0.62
$K = 2$	1.12/1.85 \approx 0.61	2.23/3.05 \approx 0.73
$K = 3$	1.27/2.11 \approx 0.60	3.40/4.48 \approx 0.76

is not surprising since, as we said, both models theoretically correspond to the geometric graph.

For $K = 2$, the 2-Robust components are those of the 1-Robust pruned of the leaves. It is therefore logical that percolation occurs at the same time ($\lambda_c^{2\text{-Robust}} = \lambda_c^{1\text{-Robust}}$ and $\lambda_{\min}^{2\text{-Robust}} \approx \lambda_{\min}^{1\text{-Robust}}$) but $\lambda_{\max}^{2\text{-Robust}} \gtrsim \lambda_{\max}^{1\text{-Robust}}$ to compensate for all the leaves (not a null proportion) that have been discarded from the giant component. Consequently $v^{2\text{-Robust}} \lesssim v^{1\text{-Robust}}$.

Unlike the 3-Robust components, which remain fairly close to the 2- and 1-Robust components, the 2- and 3-polyhedra start to appear much later: the constraints being much stronger (both on the points and on the links).

For reasons analogous to the discrete case studied in [16], the percolation rate of K -polyhedra is increasing with K and getting closer to that of a perfect classifier (in the sense of an instantaneous percolation $v \rightarrow 1$):

$$v^{1\text{-poly}} = 0.62 \ll v^{3\text{-poly}} = 0.76.$$

Results in \mathbb{R}^3 . We will go much faster when analysing the results for the dimension $d = 3$. Everything we have said about the dimension $d = 2$ remains valid. The curves have exactly the same shape. The interesting thing we noted is that with an extra degree of freedom, it is more difficult to “pick up” almost all the points of the point cloud \mathcal{X}_n , and the percolation rates are consequently smaller. This phenomenon is yet another reason for favoring the use of more complex structures such as K -polyhedra.

Table 2. Percolation rate v^K in \mathbb{R}^3 for K -polyhedra and K -Robust Single-Linkage components. $K \in \{1, 2, 3\}$, $v^K := \frac{\lambda_{\min}^K}{\lambda_{\max}^K}$ with $\varepsilon = 3\%$.

Clustering	K -Robust Single-Linkage	K -polyhedra
$K = 1$	0.42/1.05 = 0.40	0.41/1.05 \approx 0.39
$K = 2$	0.42/1.26 \approx 0.33	1.26/2.36 \approx 0.53
$K = 3$	0.53/1.60 \approx 0.33	2.37/3.81 \approx 0.62

We took the parameters $l = 15$, $\delta = 5$, $\lambda_{\text{sup}} = 6$, $m = 32$ and obtained the percolation rates listed in Tab. 2. It illustrates the great improvement w.r.t. 3-polyhedra compared with the best K -Robust percolation rate (which is once

again that of geometric graph, with $K = 1$):

$$v^{3\text{-poly}} = 0.62 \gg v^{1\text{-Robust}} = 0.40.$$

4 Conclusion and Discussion

In this article, we have tackled the vast problem of clustering *via* a density-based approach. More specifically, we focused on a family of clustering algorithms constructing (hyper-)graphs on the data depending on a scale parameter. Our goal is to extract the best possible *high-density clusters*.

We have shown that the mathematical phenomenon behind the success of these algorithms is the *percolation*. Depending on how these (hyper-)graphs are constructed and how their connected components are defined, the speed at which these objects ‘percolate’ will change.

Armed with the *percolation rate*, an index for measuring the performance of clustering algorithms we defined last year in Complex Networks [18], we showed that the natural way to generalize Single-Linkage with our K -polyhedra produces better theoretical result than the State-of-the-Art, the K -Robust Single-Linkage components (used *e.g.* by HDBSCAN).

In the future, we intend to study the phenomenon of percolation on other data than point clouds in Euclidean space. Similar results apply, for example, to the case of community detection in a Stochastic Block Model [31].

In addition, the major challenge of using hypergraphs is the computation complexity. Finding optimized algorithms/heuristics is a very important avenue.

Finally, working with Čech complexes in high-dimensional Euclidean space \mathbb{R}^d suffers from the curse of dimensionality. For example, the best actual algorithm MiniBall(x_1, x_2, x_3, \dots) for extracting the center and the radius of the smallest ball encompassing a set of $K + 1$ points $\{x_1, x_2, x_3, \dots\}$ in \mathbb{R}^d is the one proposed by WELZL [34]. This algorithm is at the heart of the Čech complex computation. Its complexity is linear in K . Unfortunately, this complexity explodes with dimension d : into $O(d^2 d!)$ [34]. In practice, even the heuristic proposed by WELZL only works up to dimension $d = 20$. Embeddings in order to reduce space dimension (in Euclidean space [23] or in *hyperbolic spaces* [19], space with good theoretical properties for networks), could thus be an interesting direction for future research.

References

1. Biau, G., Devroye, L.: Lectures on the Nearest Neighbor Method, vol. 246. Springer (2015). DOI 10.1007/978-3-319-25388-6
2. Bobrowski, O., Kahle, M.: Topology of rand. geom. complexes: a survey. Journal of Appl. and Comput. Top. **1**, 331–364 (2018). DOI 10.1007/s41468-017-0010-0
3. Bollobás, B., Riordan, O.: Percolation. Cambridge University Press (2006). DOI 10.1017/CBO9781139167383

4. Broadbent, S.R., Hammersley, J.M.: Percolation processes: I. crystals and mazes. *Mathematical Proceedings of the Cambridge Philosophical Society* **53**(3), 629–641 (1957). DOI 10.1017/S0305004100032680
5. Campello, R.J.G.B., Kröger, P., Sander, J., Zimek, A.: Density-based clustering. *WIREs Data Mining and Knowl. Disc.* **10**(2) (2020). DOI 10.1002/widm.1343
6. Campello, R.J.G.B., Moulavi, D., Sander, J.: Density-based clustering based on hierarchical density estimates. In: *Advances in Knowledge Discovery and Data Mining*, pp. 160–172. Springer, Berlin, Heidelberg (2013). DOI 10.1007/978-3-642-37456-2_14
7. Chaudhuri, K., Dasgupta, S.: Rates of convergence for the cluster tree. In: J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, A. Culotta (eds.) *Advances in Neural Information Processing Systems*, vol. 23. Curran Associates, Inc. (2010)
8. Duminil-Copin, H.: Sixty years of percolation. In: *Proceedings of the International Congress of Mathematicians, Rio de Janeiro*, pp. 2829–2856. World Scientific (2018). DOI 10.9999/icm2018-v4-p
9. Duminil-Copin, H., Raoufi, A., Tassion, V.: Subcritical phase of d -dimensional Poisson–Boolean percolation and its vacant set. *Annales Henri Lebesgue* **3**, 677–700 (2020)
10. Florek, K., Lukaszewicz, J., Perkal, J., Steinhaus, H., Zubrzycki, S.: Sur la liaison et la division des points d’un ensemble fini. *Colloquium Mathematicum* **2**(3-4), 282–285 (1951). DOI 10.4064/cm-2-3-4-282-285
11. Gower, J.C., Ross, G.J.S.: Minimum spanning trees and single linkage cluster analysis. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **18**(1), 54–64 (1969). DOI 10.2307/2346439
12. Hartigan, J.: Distribution problems in clustering. In: J. Van Ryzin (ed.) *Classification and Clustering*, pp. 45–71. Academic Press (1977). DOI 10.1016/B978-0-12-714250-0.50007-3
13. Hartigan, J.A.: *Clustering Algorithms*. John Wiley & Sons, Inc. (1975)
14. Hartigan, J.A.: Consistency of single linkage for high-density clusters. *J. of the Am. Stat. Ass.* **76**(374), 388–394 (1981). DOI 10.1080/01621459.1981.10477658
15. Hartigan, J.A.: Estimation of a convex density contour in two dim. *J. of the Amer. Stat. Ass.* **82**(397), 267–270 (1987). URL <http://www.jstor.org/stable/2289162>
16. Hauseux, L.: How can we theoretically measure the performance of density-based clustering algorithms? In: *ACM SIGMETRICS 2024 Student Research Competition*. Venice, Italy (2024). Second prize
17. Hauseux, L., Avrachenkov, K., Zerubia, J.: Benefits of hypergraphs for density-based clustering. In: *32nd European Signal Processing Conference (EUSIPCO)*. Lyon, France (2024)
18. Hauseux, L., Avrachenkov, K., Zerubia, J.: Graph based approach for galaxy filament extraction. In: *Complex Networks & Their Applications XII*, pp. 384–396. Springer (2024). DOI 10.1007/978-3-031-53472-0_32
19. Kovács, B., Kojaku, S., Palla, G., Fortunato, S.: Iterative embedding and reweighting of complex networks reveals community structure (2024). DOI 10.48550/arXiv.2402.10813
20. Last, G., Penrose, M., Zuyev, S.: On the capacity functional of the infinite cluster of a Boolean model. *The Annals of Applied Probability* **27**(3), 1678 – 1701 (2017). DOI 10.1214/16-AAP1241
21. Last, G., Penrose, M., Zuyev, S.: Corrections: On the capacity functional of the infinite cluster of a Boolean model. *The Annals of Applied Probability* **34**(3), 3370 – 3374 (2024). DOI 10.1214/23-AAP2043

22. McInnes, L., Healy, J.: Accelerated hierarchical density based clustering. In: IEEE International Conference on Data Mining Workshops (ICDMW), pp. 33–42 (2017). DOI 10.1109/ICDMW.2017.12
23. McInnes, L., Healy, J., Melville, J.: UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction (2020). DOI 10.48550/arXiv.2109.02508
24. Meester, R., Roy, R.: Continuum Percolation. Cambridge Tracts in Mathematics. Cambridge University Press (1996). DOI 10.1017/CBO9780511895357
25. Mertens, S.: Exact site-percolation probability on the square lattice. *J. of Physics A: Mathematical and Theoretical* **55**(33) (2022). DOI 10.1088/1751-8121/ac4195
26. Munkres, J.R.: Elements of Algebraic Topology. CRC Press (1984). DOI 10.1201/9780429493911
27. Müller, D.W., Sawitzki, G.: Excess mass estimates and tests for multimodality. *Journal of the American Statistical Association* **86**(415), 738–746 (1991)
28. Penrose, M.: Random Geometric Graphs, vol. 5. Oxford University Press (2003). DOI 10.1093/acprof:oso/9780198506263.001.0001
29. Quintanilla, J., Torquato, S., Ziff, R.: Efficient measurement of the percolation threshold for fully penetrable discs. *Journal of Physics A* **33**(42), L399–L407 (2000). DOI 10.1088/0305-4470/33/42/104
30. Rolle, A., Scoccola, L.: Stable and consistent density-based clustering (2023). DOI 10.48550/arXiv.2005.09048
31. Schawe, H., Hartmann, A.K.: Large deviations of connect. comp. in the stoch. block model. *Phys. Rev. E* **102**, 052,108 (2020). DOI 10.1103/PhysRevE.102.052108
32. Tobin, J., Zhang, M.: A theoretical analysis of density peaks clustering and the component-wise peak-finding algorithm. *IEEE Trans. on Patt. Anal. and Machine Intell.* **46**(2), 1109–1120 (2024). DOI 10.1109/TPAMI.2023.3327471
33. Vinay Kumar, B., Kashyap, N., Yogeshwaran, D.: An analysis of probabilistic forwarding of coded packets on random geometric graphs. *Performance Evaluation* **160**, 102,343 (2023). DOI 10.1016/j.peva.2023.102343
34. Welzl, E.: Smallest enclosing disks (balls and ellipsoids). In: H. Maurer (ed.) *New Results and New Trends in Computer Science*, pp. 359–370. Springer Berlin Heidelberg, Berlin, Heidelberg (1991). DOI 10.1007/BFb0038202
35. Wishart, D.: Mode analysis: a generalization of nearest neighbour which reduces chaining effects (with discussion). *Numerical taxonomy* pp. 282–311 (1969)
36. Xu, W., Wang, J., Hu, H., Deng, Y.: Critical polynomials in the nonplanar and continuum percolation models. *Phys. Rev.* **103**, 1–11 (2021). DOI 10.1103/PhysRevE.103.022127