Persistence-Based Clustering in Riemannian Manifolds

– Machine Learning and Data Mining project report –

Master in Computer Science

Minh Hoang DAO

June 4, 2023

Minh Hoang DAO
daohoang.hust@gmail.com
Student ID: 22213844

Supervisor: Prof. Engelbert Mephu Nguifo

Institut Supérieur d'Informatique, de Modélisation et de leurs Applications
Université Clermont Auvergne
1 Rue de la Chebarde
63178 Aubière
France

Contents

1 Introduction

Clustering problem is one of the most important problem in machine learning. There are many strategies for solving clustering, one of them is hierarchical clustering. The idea is simple, instead of strictly partitioning points into clusters, we have a sequence of nested clusters with decreasing or increasing resolution, gradually merging or separating points in clusters. The more popular way is going from bottom-up, which means regularly merging clusters together, this is called agglomerative hierarchical clustering, but the question is, how to merge?

To do so, we need define a quantity to evaluate how clusters should be merged. We can assume we have points embedded in a metric space, which means we can compute the distance between any pair of points. These distance somewhat define the similarity of points, we call it proximity matrix with all pair's values filled in. Next, in order to merge points or clusters in general, we need to define how to choose clusters for merging and how to compute new relative distances of new merged cluster to others, or how to collapse chosen clusters in proximity matrix. For choosing clusters to merge, we can naturally choose the smallest value in the proximity matrix, but for the new relative distances, we have a lot of ways to compute such as min, max, group average, centroids-based... All of them have their own pros and cons and may perform very differently in each scenario, and one of the most severe issue is they produce bad results on highly non-convex clusters.

Each way above is likes locally seeing the problem in one direction view only, not the global inherent structure of data points. Inspired from this, we can think of using a mathematics domain called algebraic topology to study this problem. This is very reasonable because algebraic topology is a field of studying the inherent properties of topology structures, which are similar to data points spaces, discrete versions of topology objects.

The study of this research topic is about the paper of Chazal et al. (2011), which used homology property of simplicial complex to build a persistence homology clustering algorithm. This is a inherent structure algorithm for hierarchical dataset, not depending on a predefined metric for merging like usual hierarchial algorithms thanks to invariant algebra properties.

Some images and explanation is also come from materials of chapter 10 in the book of Carter (2020).

2 Homology and persistent homology

Homology provides a method to associate a sequence of algebraic structures. The structures computed by homology are algebraic invariants, meaning they depend only on topological properties of the space, and remain constant under allowable transformations, e.g., stretching or bending.

More precisely, homology associates to a space X a sequence of abelian groups $H_0(X), H_1(X), H_2(X), ...$, such that $H_d(X)$ gives a measure of the number of "d-dimensional holes" in X. Because of remaining constant under continuous deformations of the space, it is sufficient to compute the homology of a simplicial complex which has the same shape as X.

A simplicial complex is a collection of points, line segments, triangles, tetrahedra, and "higher-dimensional tetrahedra," called *simplices*, together with data specifying how these spaces are attached. Simplicial complexes are useful in data analysis because a simplicial complex built on top of a finite dataset that can be computed on computer.

One common method of defining a simplicial complex from a metric space X is the *Vietoris–Rips*. The Vietoris–Rips simplicial complex with vertex set X and scale parameter $r > 0$, denoted $VR(X; r)$, contains a simplex $\{v_0, ..., v_d\}$ whenever its diameter is less than r, i.e., $diam(\{v_0, ..., v_d\}) < r$. If we consider an increasing sequence of scale parameters $r_1 \leq r_2 \leq ... \leq r_{m-1} \leq r_m$, then we obtain an increasing sequence of simplicial complexes as in Fig 1.

$$VR(X; r_1) \subseteq VR(X; r_2) \subseteq ... \subseteq VR(X; r_{m-1}) \subseteq VR(X; r_m).$$

Persistent homology will give us a language to describe not only the holes in $VR(X; r_j)$ at each scale r_j , but also how the holes in $VR(X; r_j)$ relate to those in $VR(X; r_{j+1})$, in particular, which holes at scale r_j die at scale r_{j+1}, and which holes remain or persist from scale r_j to scale r_{j+1}.

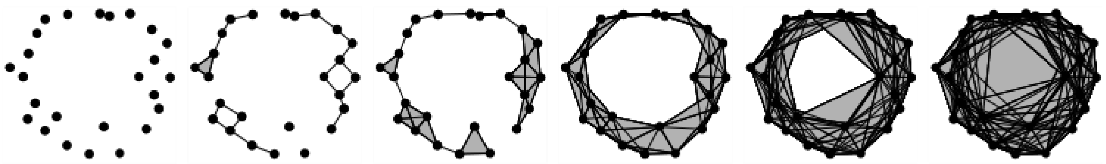

Figure 1: A dataset and its Vietoris–Rips complexes at six different choices of scale.

The *i-dimensional persistent homology* of this increasing sequence of spaces can be represented as a set of intervals (Fig 2), referred to as the persistence barcode. The information in a persistence barcode can equivalently be presented as a persistence diagram. Indeed, in a persistence diagram, each interval I with birth-time b and death-time d is represented as the point (b, d) in the plane $\mathbb{R}^2$ as in Fig 3, some illustration can swap these two axes.

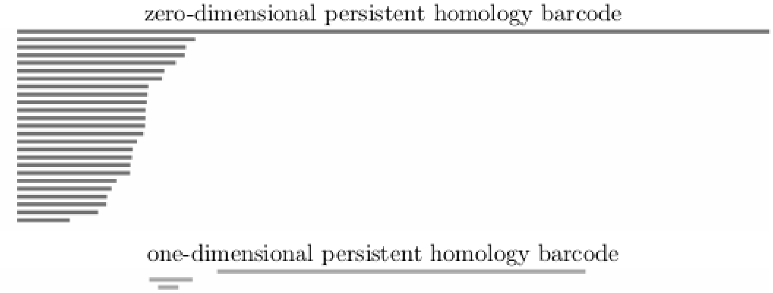

Figure 2: The zero-dimensional and one-dimensional persistent homology intervals.

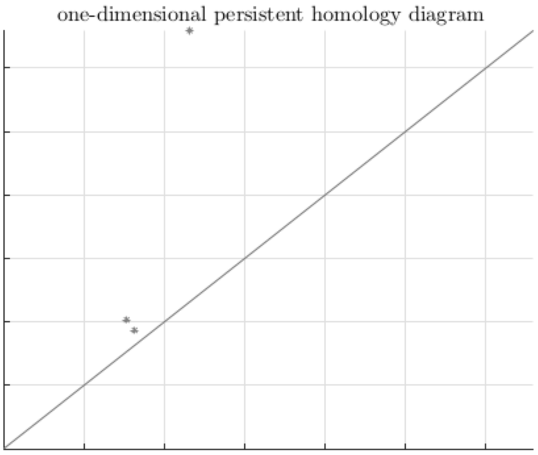

Figure 3: The one-dimensional persistent homology diagram.

3 Persistence-based clustering algorithm

This part is about the paper of Chazal et al. (2011), using the 0^{th} persistence homology group to find the number of clusters and then using a density function for constructing tree structure of data points before merging points to form the final clusters. The following part presents the main steps of the algorithm with the illustration as in Fig 4 and 5.

1. Initialize the neighborhood graph on data points, using Vietoris–Rips complex or Delaunay graphs or k-nearest neighbor (Fig 4c).

2. Use some density estimator (Gaussian kernel estimator for example) to give each vertex of the graph a value (Fig 4b).

3. Compute the persistence diagram, persistence barcode to chose the number of clusters and the merging parameter τ (Fig 5b).

4. Compute the initial clusters by using the estimated density to connect each vertex to its neighbor. The result is a spanning forest over the graph with pseudo-gradient edges (Fig 5a).

5. Merge trees due to the density value of their root to form the final clusters (Fig 5c).

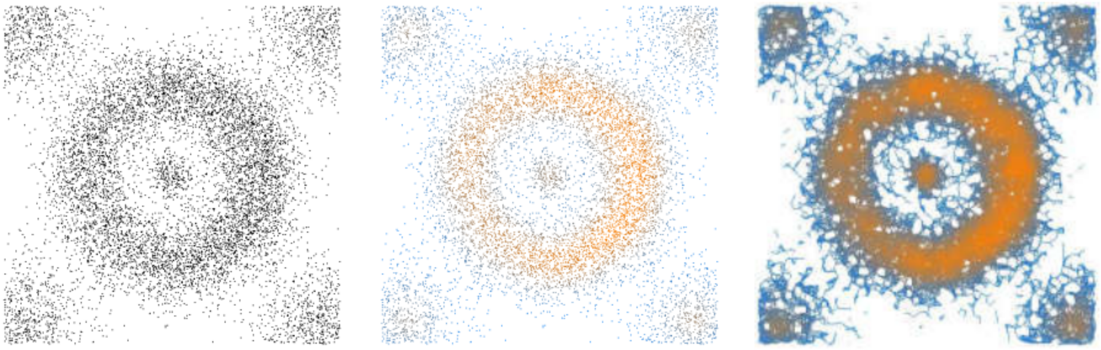

Figure 4: (Left-to-right) a) Input point cloud. b) Approximated density function. c) Rips graph of the point cloud.

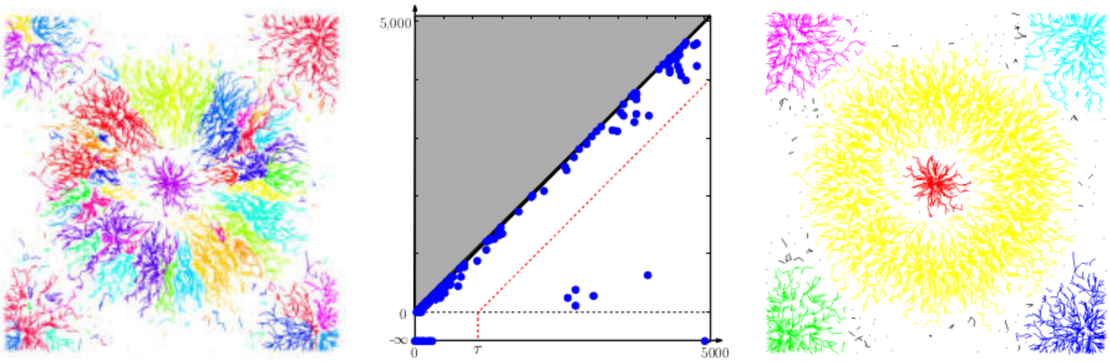

Figure 5: (Left-to-right) a) Initial clusters before merging. b) Persistent diagram. c) Final clusters after merging.

4 Experimental result

The experiment used TdaToolbox[1] which based on GUDHI library[2] to compute, visualize persistence diagram, barcode and also run the algorithm. The toy data set have the form of two twisted spirals with added noise as in Figure 6.

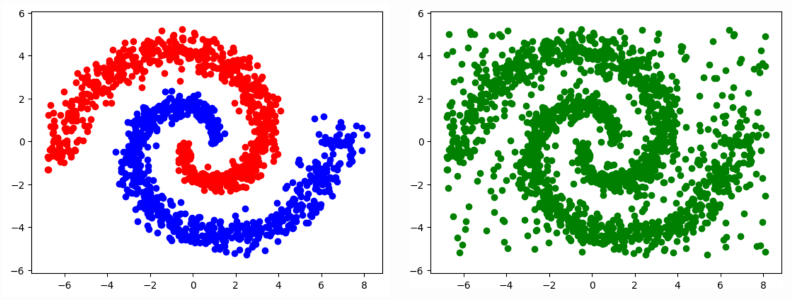

Figure 6: (Left) Two separated spirals. (Right) Noise added data.

The persistence diagram and bardcode is shown in Figure 7. We can see that there are two points in the diagram are much higher from the diagonal and their corresponding lines in the barcode that are very longer than others. One of two lines is shorter because of the noise that causes clusters easier to be connected in the Vietoris–Rips complex. The final result is shown in Figure 8.

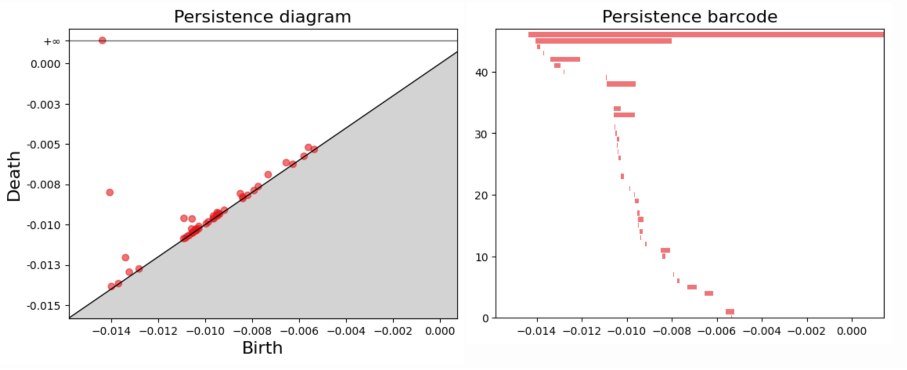

Figure 7: (Left) Persistence diagram. (Right) Persistence barcode.

5 Conclusion

The research topic is about exploiting inherent properties of input data in clustering problem, which uses algebraic invariants in algebraic topology. This approach

[1] https://github.com/merylldindin/TdaToolbox.

[2] https://gudhi.inria.fr/.

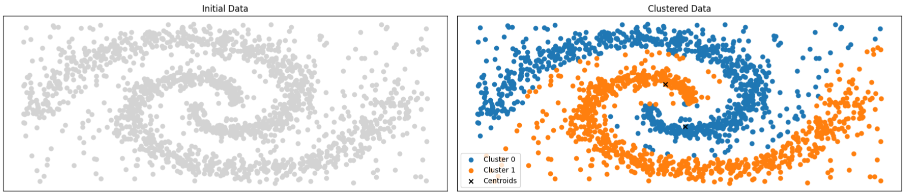

Figure 8: The result of the algorithm.

somehow overcomes severe issues of hierarchical clustering algorithm, which contain not dealing with highly non-convex clusters and only partially viewing data with some corresponding metrics. Because algebraic invariants of topology object are global properties, so the approach seems very robust to noise, in fact it is, this is proved in the paper of Chazal et al. (2011). The way algebraic topology is applied to Topological Data Analysis is quite new and very promising. The experiment uses very popular framework named GUDHI and showed the efficiency in dealing with noisy, twisted, non-convex toy data set.

References

Carter, N. 2020. *Data Science for Mathematicians.* 441. Routledge.

Chazal, F., L. Guibas, S. Oudot, and P. Skraba. 2011. “Persistence-Based Clustering in Riemannian Manifolds.” *Journal of the ACM,* 60.