Foundations of Geometric Methods in Data Analysis
Project10: Geodesic filtrations for segmentation

DARGIER Antoine   RAMAMONJY Johary

May 7, 2023

1 Introduction

1.1 Topological Data Analysis

Topological data analysis (TDA) is an emerging field of data analysis that uses topological methods to analyze and extract information from complex and high-dimensional data sets. It involves studying the shape, structure, and connectivity of data, rather than just focusing on its individual data points. TDA provides a powerful and flexible toolset for exploring and understanding the topological features of data, such as the number of connected components, holes, loops, and voids, as well as their relationships and interactions. It is based on the mathematical theory of topology, which deals with the properties of space that are preserved under continuous transformations such as stretching, bending, and twisting. TDA has a wide range of applications in various fields such as biology, neuroscience, physics, economics, social sciences, and computer science. It can be used for data visualization, pattern recognition, clustering, classification, dimensionality reduction, and anomaly detection, among other tasks. Some common techniques used in TDA include persistent homology, which measures the persistence of topological features across different scales, and mapper, which constructs a simplicial complex from the data and captures its topological structure. TDA has the potential to unlock new insights and discoveries in complex data sets and is an active area of research and development.

1.2 The project

The main objective of this project is to develop and implement geodesic filtrations as a method to address the *segmentation problem* in 3D shape analysis. This problem involves the classification of points within 3D shapes, which is a fundamental task in a variety of applications such as computer graphics, computer vision, and geometric modeling. The proposed approach involves filtering the 3D shapes locally using geodesic distances to basepoints, which are points on the surface of the shape that are selected as the starting points for the filtration process. We have based our study on this paper[1]. The paper discusses the use of persistence diagrams to analyze the local structure of discrete geometric objects, such as meshes and point clouds, and how this can be applied to problems in computer graphics. While existing methods rely on either very local or global analyses, the authors propose a new descriptor, which is the state of the shape as seen from a particular point. They also show how this kind of descriptor can be transformed into vectors, which can then be used to train classification models to distinguish shapes or recognise parts.

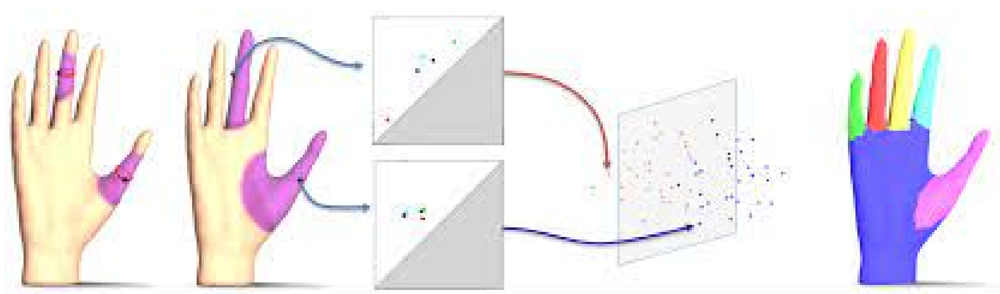


Figure 1: Computation of persistent diagrams from two different basepoints, and shape classification

Overall, this project presents a promising approach to address the segmentation problem in 3D shape analysis, which has numerous practical applications in various fields. The proposed method has the potential to improve the accuracy and efficiency of shape analysis tasks and can be extended to handle more complex shapes and datasets.

2 Geodesic Distances

2.1 Computation

To construct the persistence diagrams, we will construct a metric that we will use as filtration on the Simplex Tree. To do this, we will implement an algorithm that approximates geodesic distances on a 3D shape triangulation using the shortest distances on the corresponding graph using Dijkstra's method. Geodesic distances refer to the shortest distance between two points on a curved surface or manifold, such as a sphere, a torus, or a more complex shape. In contrast to Euclidean distances, which are measured in a straight line between two points, geodesic distances take into account the curvature and topology of the surface. We will start by constructing a graph, where each node is a point of our shape, which will be connected to its neighbours according to the faces of the triangulation. We associate the weight to this edge the Euclidean distance between these points because on these small triangulations, we will consider that the shape is flat. Then, we just have to go through the graph adding the distances to obtain all the necessary geodesic distances.

To obtain the distance, we use a graph traversal algorithm for finding the shortest path: Dijkstra's algorithm. Unlike other algorithms, Dijkstra's algorithm is based on a breadth-first approach. This algorithm is of polynomial complexity. More precisely, for n vertices and α arcs, the time is in $O((\alpha + n)\log n)$. Here is the pseudo-code for a non-oriented weighted graph:

Algorithm 1 Dijkstra's algorithm for a weighted non-oriented graph

Require: A weighted non-oriented graph $G = (V, E)$, a source vertex s
Ensure: The shortest path from s to all other vertices in G
  Initialize a set S of visited vertices and a set Q of unvisited vertices
  Set the distance to the source vertex s to 0 and all other distances to infinity
  Set the previous vertex of all vertices to null
  while Q is not empty **do**
    Choose the unvisited vertex u with the smallest distance to s
    Add u to S
    for each neighbor v of u **do**
      Calculate the distance from s to v passing through u
      if the calculated distance is less than the current distance to v **then**
        Update the distance to v
        Set the previous vertex of v to u =0

2.2 Results

Finally, for different shapes, we were able to create these graphs, and then calculate the approximate geodesic distances. So, for the three shapes we took as examples, we have on the left the shape with the different classified parts, and on the right the figure with the distances from a randomly chosen base point, represented in black.

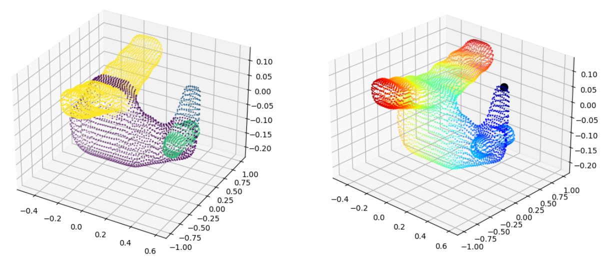


Figure 2: Left: segmented shapes, Right: geodesic distances from basepoint in black

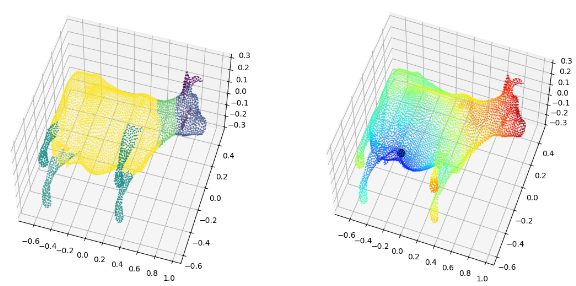


Figure 3: Left: segmented shapes, Right: geodesic distances from basepoint in black

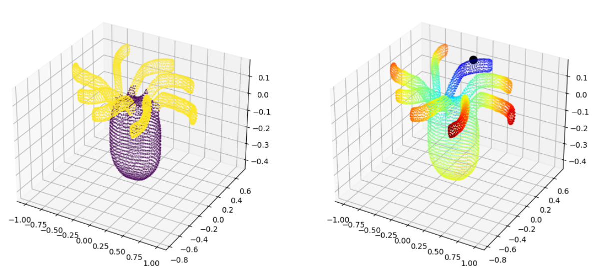


Figure 4: Left: segmented shapes, Right: geodesic distances from basepoint in black

2.3 Interests

The interest in using geodesic distances as filtration in topological data analysis for shape segmentation and computing persistent diagrams lies in their ability to capture the intrinsic geometry and topology of 3D shapes. Geodesic distances are a natural choice for filtration because they reflect the underlying shape of the surface, taking into account its intrinsic curvature and topology. This makes them particularly useful for distinguishing between different regions of a shape and identifying meaningful boundaries between them.

Moreover, geodesic distances are invariant to solid transformations such as translation, rotation, and scaling, making them robust to noise and deformation. This allows for the accurate segmentation and identification of shapes, even in the presence of noise or partial occlusion.

3 Point signatures

Filtration is an important step in the computation of persistent homology, including the persistent diagram using Gudhi. Persistent homology is a mathematical tool used to analyze the topological features of a dataset at multiple scales. The idea behind filtration is to transform the dataset into a series of simplicial complexes by gradually increasing the level of detail or granularity. In other words, filtration is a way of gradually building up the structure of the dataset, starting from a very coarse level and gradually becoming more detailed. By doing this, we can study the evolution of the topological features of the dataset as we move through the filtration, and this information is captured in the persistent diagram. The persistent diagram is a visual representation of the birth and death times of the topological features of the dataset, which allows us to analyze and compare the persistence of these features at different scales. Therefore, filtration is an essential step in the computation of the persistent diagram using Gudhi. Without filtration, we would not be able to analyze the persistence of the topological features of the dataset, and the resulting persistent diagram would be meaningless.

So here, the filtration process involves constructing a hierarchy of subsets of points based on their distance to the basepoints. These subsets are called the geodesic balls and their union covers the entire shape. The geodesic balls are filtered based on their intrinsic properties, such as the distribution of points within them, which results in a sequence of nested sets. The filtration process generates a signature for each point based on its membership in each set of the filtration sequence. If we use the distance function directly, we will always have a pair $[0, +inf]$ because the filtration is composed of a geodesic ball with a single connected component which is only growing, so there is nothing topologically interesting. On the other hand, if we use the opposite, we will first take the most distant points (in the sense of geodesic distance) which can form several related components.

The resulting signatures provide a concise and informative representation of the local geometry of the shape, capturing both global and local features. These signatures can be used for various tasks such as shape classification, segmentation, and retrieval. Furthermore, as seen before, using geodesic distances ensures that the signatures are robust to noise and deformation, making them suitable for real-world applications.

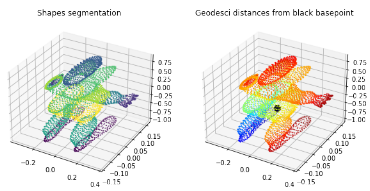
Here is the pseudo-code we used to assign the opposite of geodesic distances to each vertex, and then compute the persistence diagram:

Algorithm 2 Point signatures computation

Require: Vertices, faces, basepoints
Ensure: The persistence diagram for each basepoint
  Create the graph from vertices and faces
  for Each basepoint **do**
    Create a Gudhi's SimplexTree
    Compute the distances dictionary from the basepoint
    for Each vertex **do**
      $Filtration[vertex] \leftarrow -distances[vertex]$
    end for
    Make filtration non decreasing
    Create and store persistence diagram
  end for=0

4 Stability Theorem

Here we will look at the stability of this signature. As the article says, "the main advantage of considering topological signatures is that they enjoy stability properties, meaning that the difference between two signatures cannot be too large if the signatures are computed from nearby points or on nearby shapes. This stability is a key feature in applications." Indeed, the construction of the signature is done in two steps. The first is the construction of the persistence diagrams (in homology dimension 0), which are stable if we place ourselves at very close points or for close shapes. Then there is the stability of the vectors that we use from these persistence diagrams. To verify this, we carried out two experiments. Firstly, we drew the persistence diagrams for two close shapes, placing the basepoint at the same location. Here, we chose human shapes and placed our basepoint at the end of the left foot. As you can see, the persistence diagrams are very close.

[Human shape 1] 

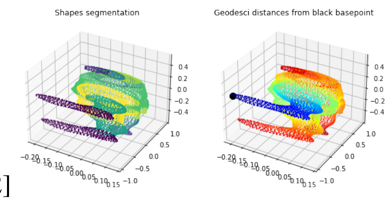
[Human shape 2] 

Figure 5: The two shapes on which we will compute the persistent diagrams, with the black basepoints

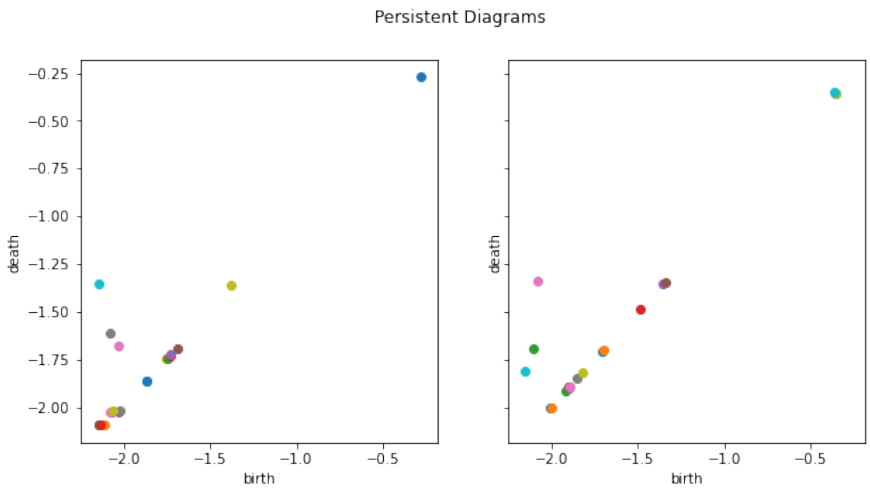


Figure 6: Persistent diagrams for two close shapes with same basepoint

For a second experiment, we kept only one shape and chose two nearby basepoints. We kept the human shape and took two nearby points on the left foot. The results confirm the stability theorem.

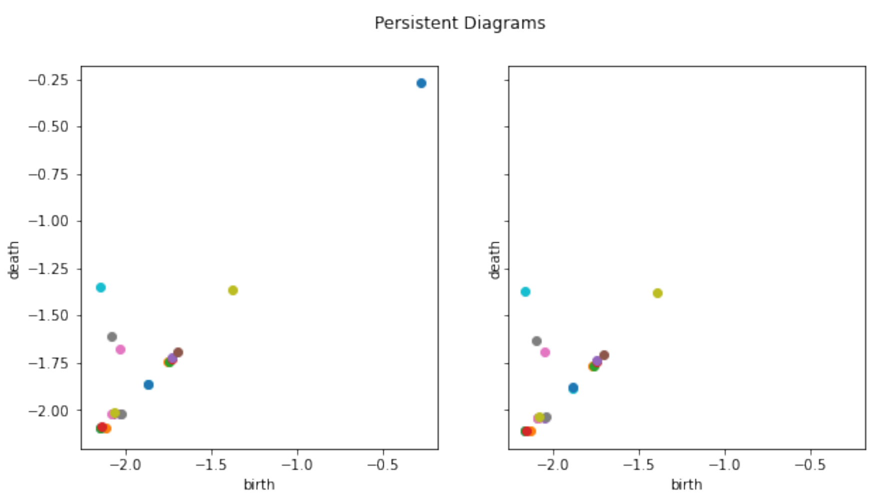


Figure 7: Persistent diagrams for two close basepoints on one shape

5 Statistical tests

We then design statistical tests to distinguish a shape versus another from their family of persistence diagrams. To do this, we will use the Maximum Mean Discrepancy defined in [2]. It is a method to compare two probability measures. It is based on the fact that two probability measures p and q are equal if and only if for all measurable bounded function f, $\mathbb{E}_p [f(X)] = \mathbb{E}_q [f(Y)]$ with p (resp. q) is the probability law of X (resp. Y). Given a set of bounded functions F, the Maximum Mean Discrepancy is defined by :

$$\text{MMD}[F, p, q] = \sup_{f \in F} \mathbb{E}(f(X)) - \mathbb{E}(f(Y))$$

Given $x_1, ..., x_m$, and $y_1, ..., y_n$ samples of X, and Y, we obtained the empirical estimate of the MMD :

$$\text{MMD}[F, p, q] = \sup_{f \in F} \left(\frac{1}{m} \sum_{i=1}^{m} f(x_i) - \frac{1}{n} \sum_{i=1}^{n} f(y_i) \right)$$

Applied to persistence landscapes V, the MMD between two diagram S_1 and S_2 is :

$$\text{MMD}(S_1, s_2) = \sup_{f \in F} \left\| \frac{1}{|S_1|} \sum_{x \in S_1} V(f(S_1)) - \frac{1}{|S_2|} \sum_{x \in S_2} V(f(S_2)) \right\|$$

We apply this with permutation tests.

In our project, we tried to compare human and ant diagrams. To do this, we choose randomly two human diagrams and two ant diagrams. Than, we removed the infinite points. To compare two diagrams, we calculate MMD on their persistence landscapes. Then, we perform 1000 permutation tests, where we shuffle the points of the two diagrams and create two random shuffled diagrams. We compute the MMD on this new diagrams and calculate the p-value, which is here the proportition of permuted diagram that have a higher MMD than on the original diagram. The threshold of the p-value is set to 0.05. Under this threshold, the persistence diagrams are considered to be different, and above, they are considered to represent the same object.

We got the right results on comparing the two randomly chosen human and ant diagrams : the two humans diagrams were classified as same object as well as for the two ants diagrams, and when comparing a human diagram with an ant diagram, the algorithm was able to differentiate them. However, it could have been better two try our algorithm on more diagrams from human and ant, but it would have been more time and resources consuming. Moreover, we could also compare other diagrams. When experimenting, the algorithm could not differentiate human diagrams with hand diagrams.

6 Segmentation models

Finally, we have seen that these persistence diagrams make it possible to distinguish the different parts of a shape. We will therefore use them to answer the ”segmentation problem”, i.e. to classify the different parts of an element. For that, we begin by computing all the persistence diagrams (in homology dimension 0) associated with the sublevel sets of the geodesic distances with as basepoints all points in the shapes and retrieve their corresponding labels. This step took about 5 minutes to complete for a shape. To do this, we can use a function that sends persistence diagrams to a Hilbert space, or we can use a symmetric kernel function. This kind of function already exists in the Gudhi library, which will be very useful. We can use different persistence diagrams representations, such as persistence landscapes or persistence images. Both persistence landscapes and persistence images are tools used in topological data analysis to visualize and analyze the persistence homology of a given data set. Persistence landscapes are constructed by taking the sum of smooth basis functions and are typically visualized as a continuous function, with the x-axis representing time and the y-axis representing amplitude. On the other hand, persistence images are constructed by mapping the persistence diagrams onto a grid of pixels and are visualized as a two-dimensional image, with each pixel representing a particular topological feature at a given persistence level.

Here are the different steps we can use and play with to find the best classification model:

- a separator: for selecting the finite points and remove infinity values
- a scaler: to rescale the value if needed
- a TDA: the vectorization method to transform the PDs
- a model: the model used for the classification

We have tried many possibilities for the separation into train and test sets. In addition, we used cross-validation and the GridSearch function to search for the best hyperparameters. However, using all the points of the shapes (about 5,000), the training was extremely time-consuming (in several hours), especially for the SVM model which is often the longest. We, therefore, reduced the train set size to 5%, and kept a RandomForest and a kNN as classification models. Given the stability properties of our signature, these models seem to be fully appropriate for a good classification.

We thus obtained excellent performance, with an accuracy of 92% on the test set with this model and hyperparameters:

```
{'Estimator': RandomForestClassifier(), 'Scaler__use': True, 'Separator__use': True, 'TDA': Silhouette(), 'TDA__resolution': 100}
Train accuracy = 1.0
Test accuracy  = 0.9228187919463087
```

Figure 8: Model hyperparameters and accuracy

Our model is thus able to recognize and classify elements on shapes!

7 Conclusion

In the first part of the project, we explored the use of geodesic filtrations for shape segmentation. To do this, we first created geodesic distances on shapes, which measure the shortest path between any two points on the shape's surface. This distance measure is invariant to translations, rotations, and scaling, making it ideal for comparing shapes. Next, we used the geodesic distances as a filtration to compute persistence diagrams. Persistence diagrams are a way to capture the topological features of a shape at different spatial scales. They represent the birth and death times of connected components, holes, or voids that appear and disappear as the filtration parameter varies. By studying these persistence diagrams, we gained insights into the topological properties of the shapes and the relationships between their different parts. To evaluate the stability of our method, we used the stability theorem, which states that small changes in the shape or the filtration parameter should result in only small changes in the persistence diagrams. We tested the theorem by computing the PDs for similar shapes or close basepoints and comparing the resulting persistence diagrams to the original ones. The results showed that our method was robust to these perturbations and produced stable persistence diagrams. We designed statistical tests to distinguish between different shapes based on their persistence diagrams. We used techniques from statistical hypothesis testing to compute p-values and confidence intervals, which allowed us to determine whether two shapes were significantly different or not. Finally, we have used different methods to vectorise persistence diagrams, and have used them to train machine learning models. With excellent accuracy, we are able to recognise and classify different parts of a human body, such as arms, legs or head.

There are several ways in which the approach we developed in this project can be improved. One possible area for improvement is the choice of distance measure. While geodesic distance is a robust and accurate measure of shape similarity, it can be computationally expensive to compute, especially for large or complex shapes. One potential solution is to use other distance measures, such as the heat kernel signature, which can be computed more efficiently and provide similar levels of accuracy. Another area for improvement is the use of machine learning to analyze and classify shapes. While we achieved good results using vectorization tools and machine learning algorithms, there is still room for improvement in terms of optimizing the architecture, hyperparameters, or feature extraction methods. For example, we could explore the use of deep learning models, such as convolutional neural networks, to extract more complex and robust features from the persistence diagrams. In addition, we could investigate other techniques for shape segmentation and analysis, such as the use of deep learning models for unsupervised learning or the incorporation of other types of topological features, such as Betti numbers or persistent homology invariants.

The methods and models developed in this project have many potential applications and benefits in various fields. In medical imaging, for example, shape analysis can be used to detect and diagnose diseases, such as tumors or lesions, by comparing the shapes of healthy and diseased organs. By leveraging the topological properties of shapes, we can extract meaningful features that enable accurate diagnosis and treatment planning. In robotics, shape analysis can be used to improve the perception and manipulation of objects by robots. By analyzing the shapes of objects, robots can better understand their properties and use this information to interact with them more efficiently and effectively. In virtual reality, shape analysis can be used to create more realistic and immersive environments by accurately modeling the shapes of objects and scenes. This can enhance the user experience and enable more realistic simulations and training scenarios. Furthermore, the methods and models developed in this project can also benefit fields such as computer graphics, geology, geography, or physics. By leveraging the topological properties of shapes, we can gain insights into the underlying structures and properties of complex systems and develop more powerful tools for analysis and classification.

References

[1] Steve Y. Oudot Mathieu Carrière and Maks Ovsjanikov. Stable topological signatures for points on 3d shapes, 08 2015. 1

[2] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012. 6