UNIVERSITAT DE BARCELONA

Facultat de Matemàtiques
i Informàtica

GRAU DE MATEMÀTIQUES

Treball final de grau

Dimensionality reduction based on persistence entropy

Autor: Junhan Cui

Directors:	**Carles Casacuberta Vergés**
	Aina Ferrà Marcús
Realitzat a:	**Departament de Matemàtiques**
	i Informàtica
Barcelona,	**13 de juny de 2023**

Contents

Abstract

In this work we use different methods from algebraic topology, statistics, and data analysis to study a specific data set. This includes tools and analysis methods such as homology, simplicial complexes, persistent homology, bottleneck distance, Wasserstein distance, total persistence, persistence entropy, and directional hierarchical analysis. Our aim is to study a database generated during a previous neuroscience experiment by Cos et al. (2021). This database is a high-dimensional electroencephalogram (EEG) data set of recordings from 11 participants in a decision-making experiment in which three motivational states were induced by manipulating social pressure onto participants.

Our goal is to find out the intrinsic dimension of this database, that is, the number of latent variables, and look for subjects in the study population who are significantly different from the rest. This work was inspired by a paper by Ferrà et al. (2023), in which the authors present a new analytical approach using topological data analysis (TDA). Traditional dimensionality reduction methods determine how many dimensions should be retained attempting to preserve variance of the data, while topological data analysis estimates an optimal dimension by studying the data's topology. While a TDA classifier was used by Ferrà et al., in this work we use directed hierarchical analysis combined with distances between persistence diagrams and persistent entropy to assess the amount of topological variation depending on the ambient dimension.

Keywords: Algebraic topology, Persistent homology, Principal component analysis, Dimensionality reduction, Persistence entropy.

2020 Mathematics Subject Classification. 55N31, 62R40, 68T09

Chapter 1

Introduction

1.1 Problems encountered in big data analysis

With the development of computer science and the advent of the information age. The amount of data generated in everyday life has grown exponentially. We have ushered in an era known as the Big Data Society. Scholars in the last century could only use very primitive data collection methods to obtain information when conducting data analysis, such as using paper questionnaires to interview passers-by on the street. The total amount of data and data dimensions that can be obtained by this survey method is not high, because it is difficult to find millions of people to conduct a certain survey (so it will lead to a lack of data volume), and let the respondents fill in a It is also unrealistic to have a questionnaire with thousands of questions (so the data dimensionality will be very low). Therefore, the amount of data that could be used in the field of data analysis at that time was actually very scarce compared to now. In order to obtain sufficient analysis data, methods such as resampling were needed to expand the amount of data.

Although we can obtain sufficient data now, the tools and means we can use are not sufficient for how to analyze and process these data. More complex data brings more information on the one hand, and more complex data structures on the other hand. Research on how to find the results we need in the huge and complex data, or reveal the potential relationship hidden behind the data that we do not know is still a thorny problem in the field of data analysis.

A common problem that plagues researchers in big data analysis is the "curse of dimensionality", which is a term first proposed by the American applied mathematician Richard Bellman when considering optimization problems. When the dimension of (mathematical) space increases, analyzing and organizing high-dimensional spaces (often hundreds or even thousands of dimensions), encounters various problem scenarios due to the exponential increase in volume. Such dif-

ficulties are not encountered in low-dimensional spaces, such as physical spaces, which are usually only modeled in three dimensions. The reason why this happens is also obvious. Because in three dimensions, each variable is connected with at most two other variables, and in N-dimensional space, each variable is connected with at most $N-1$ variables, then when the value of N is large enough, the relationship between variables The connection between each other will be like a ball of yarn that is entangled with each other and cannot be untied. In order to solve this problem, we need to use data dimensionality reduction technology to reduce the dimensionality of data while retaining information to the greatest extent for more convenient research.

1.2 Purpose of this work

The purpose of this work is to analyze data through topology in the field of mathematics rather than statistics. Among them, our main research goal is to use topological data analysis (TDA) to accomplish the following goals:

1. Data dimension reduction and find topological latent dimension of data.
2. Find the most critical variables in the data set.
3. Outlier detection.

If you use traditional data dimensionality reduction methods, such as principal component analysis (PCA). We can only preserve the original information of the data by retaining the variance. According to the definition of PCA, we know that the more principal components we keep, the more information we can get, and this leads to a problem: how much information we need to keep is determined according to our needs. If more information is needed, then more principal components should be retained, and if less information is needed, more principal components should be removed.

In this work, we explore a new possibility, namely whether there is a range so that we can reduce the data dimension by removing variables in this range without causing obvious damage to the original data structure.

Let us take a very simple example to understand what has been said above: If we now have a data set with 100 dimensions. After PCA data dimensionality reduction it was transformed into a data set with 100 principal components. If we need to retain 80% of the variance, then we sort the value of the retained variance from large to small, and then select the first 20 principal components as variables in the new data set. If we only need to retain 60% of the variance, then we choose the first 15 principal components. In other words, if only PCA is used, there is no

objective standard to tell us that we should reduce to a specific number of principal components. We ultimately decide how many principal components to choose based on how much variance we need to preserve, and this is a very subjective decision. Subjective decisions may be wrong, because we may subjectively think that retaining 15 principal components (60% of variance) is enough for analysis, but this may not be the case.

Therefore, we need a standard that can judge how many principal components need to be retained through objective facts. If we measure informativeness not by retained variance, but by the topology of the data, a different situation may arise. For example: when we select the first 20 principal components (equivalent to retaining 80% of the variance), then we find that after removing some of the five principal components (note that these five principal components are determined according to a specific standard!), the remaining 15 principal components almost maintain the topology of the data when there are 20 principal components. In other words, these 15 specific principal components have almost the same topological data structure as the original 20 principal components! Then we can remove the five principal components that do not affect the topological data structure, and achieve data dimensionality reduction without affecting the topological data structure of the original data.

Chapter 2

Data Dimensionality Reduction

2.1 Significance of dimensionality reduction

Before we start, we need to introduce first the significance of data dimensionality reduction. High-dimensional data usually causes two major problems:

1. Curse of Dimensionality
2. Visualization of Data

Visualization of data is easy to understand, since we cannot observe directly the high-dimensional (larger than 3) space, so we need to put them in a lower-dimensional space. We briefly mentioned a headache in data analysis in the introduction before, that is, the curse of dimensionality.

The "curse of dimensionality" is a term first proposed by Bellman when considering optimization problems to describe the analysis and organization of high-dimensional spaces (usually there are hundreds or thousands of dimensions), encountering various problematic scenarios due to exponentially increasing volume. Such difficulties are not encountered in low-dimensional spaces, such as physical spaces, which are usually only modeled in three dimensions.

But in the virtual data space, the dimensionality is often extremely huge. Because we can regard a collected data with n variables as a point in an n-dimensional space, and all the collected data together constitute a set in this space, we call it "data point cloud". This dimension n can be an arbitrarily large positive integer (usually tens or hundreds in practice). Then the dimension of this data space will also be far beyond three-dimensional.

A simple example is the sparsity of the available data.

Example 2.1. Logical tables in text analysis. Logical tables can encode categorical variables with One hot encoding. One hot encoding is a commonly used encoding

method for converting categorical variables into numeric vectors with a certain length. The specific method is to create a vector, only write 1 in the category that the categorical variable is actually equal to, and write 0 in the position of the rest of the possible values of this variable. Finally, when we convert all categorical variables into vectors represented by only 0 and 1, we have obtained a very large data matrix. The elements of this matrix only contain 0 and 1 and most of the positions are 0. This is the most classic "sparse" data, because we need to use a lot of space to store meaningless 0, and this will lead to low algorithm efficiency.

Another example is the existence of redundant variables.

Example 2.2. When we want to use the values of some variables in the database to predict the values of other variables, we often need to consider a question, which variables are really related to the response variables we need to predict? For example, if we want to predict the age of some people. Then the variables that may be relevant are: income level, height, weight, frequency of hospital visits, etc. Variables that do not have a significant correlation such as: eye color, breakfast preference, etc. If we do not remove these irrelevant "redundant variables" during prediction, the values of these variables may affect the accuracy of prediction. In other words, the higher the dimension, the better. We hope to keep only the variables we need and have a positive effect on our research. Therefore, we need to remove some redundant variables to improve the accuracy and algorithm efficiency.

2.2 Latent dimension

One of the goals of this paper is to find the "latent dimension" of a data set. So here we briefly introduce the definition of latent dimension and some assumptions related to it.

First of all, we need to define the conception of "latent space".

Definition 2.3. A *latent space*, also known as a latent feature space or an embedding space, is an embedding of a set of elements in a manifold where similar elements have smaller distances in the latent space. A position in the latent space is defined by a set of latent variables resulting from the similarity between elements.

In most cases, the dimensionality of the latent space is set to be lower than that of the feature space of the data points, which means that the construction of the latent space is actually a dimensionality reduction, which can also be seen as a form of data compression.

Like the original data space, each dimension of the latent space is a "latent variable".

Definition 2.4. In statistics, latent variables, or hidden variables, latent variables, as opposed to observed variables, refer to unobservable random variables. Latent variables are variables that can only be inferred indirectly through mathematical models from other observable variables that can be directly observed or measured.

The most common latent variables are the results of linear combinations of the original variables. For example, in principal component analysis (PCA), each principal component we get can be regarded as a latent variable.

A hypothesis closely related to latent dimensions is the Manifold Hypothesis.

Definition 2.5. The *Manifold Hypothesis* assumes that many high-dimensional data sets that appear in the real world actually lie on a low-dimensional latent manifold within this high-dimensional space. That is, many data sets that initially appear to require many variables to describe can actually be described by relatively few variables, likened to the local coordinate system of the underlying manifold.

This work builds on the Manifold Hypothesis, but the topic of this work is not to explain the latent space and its latent variables, but to find the number of latent variables, the so-called "latent dimension".

2.3 Dimensionality reduction methods

In this section, we briefly introduce some common data dimensionality reduction methods. The reason for introducing them is that the data dimensionality reduction method we created in this paper is partly based on them and also draws important inspiration from them.

According to different classification standards, there are many different classifications of dimensionality reduction methods. Here we classify according to whether the dimensionality reduction method is linear or nonlinear.

Definition 2.6. *Linear dimensionality reduction* refers to a data dimensionality reduction method based on linear assumptions that may lose the nonlinear structural information inside the data.

Non-Linear dimensionality reduction refers to a data dimensionality reduction method that is not based on linear assumptions and aims to capture the internal nonlinear structure of the data.

In this work we use the linear dimensionality reduction, so we just introduce them. the most commonly used linear dimensionality reduction method is factor analysis (also known as factor method). Factor analysis provided by the British

psychologist C. E. Spearman. It refers to the study of statistical techniques for extracting common factors from variable groups. Factor analysis is actually a collection of different sub-methods.Factor analysis finds hidden representative factors (that is, latent variables) among many variables. Grouping variables with the same nature into one factor can reduce the number of variables and test the hypothesis of the relationship between variables.

The output of factor analysis is always (except LDA) n factors (where n is the dimension of the original data), and a new coordinate system composed of these n factors (essentially a rotation of the original coordinate system).

There are mainly several methods in factor analysis: Correspondence Analysis (CA), Principal Component Analysis (PCA), Multiple Correspondence Analysis (MCA) and Linear Discriminant Analysis (LDA).

CA is a method specially used to find the correlation between variables (or individuals) stored in contingency table data. PCA and MCA are similar to CA, but they are used for numerical variables and categorical variables respectively. The research method by topological data analysis used in this paper is based on PCA. So for a more detailed introduction to PCA and the differences between PCA and MCA, we will explain it in detail using mathematical language in the next section.

2.4 Principal component analysis (PCA)

The data dimensionality reduction method we created using topological data analysis (TDA) in this paper is based on Principal Component Analysis (PCA). So in this section, we will focus on using mathematical language to introduce the definition of PCA, the mathematical principle, the way to interpret, the method of use and the inspiration for the method used in this work.

Definition 2.7. In multivariate analysis, Principal Component Analysis (PCA) is a method of statistical analysis, simplifying data sets. It was first introduced by K. Pearson for non-random variables, and then H. Hotelling extended this method to the case of random vectors. It uses orthogonal transformation to linearly transform the observed values of a series of possibly related variables, thereby projecting the values of a series of linearly uncorrelated variables, which are called Principal Components. Specifically, the principal component can be viewed as a linear equation, which contains a series of linear coefficients to indicate the projection direction.

The objective of PCA is to find the isomorphic transformation from original space that keeps the adjacency relationships among variables.

Express results in a fictitious space and then find the most informative projection planes (factorial plane) in that fictitious space. And the quantity of "information" is usually measured using the sum of squared deviations or variance.

We can also not find the best projection plane, but only reduce the dimensionality of the data, and study the projection coordinates of the original high-dimensional data points in low dimensions.

First of all, we need to know that PCA is only available for numerical variables, which means that no categorical variables are allowed.

And then we need to define some basic concepts officially.

Definition 2.8. In the field of data analysis, *Inertia* refers to statistical inertia, that is, variance that one variable has.

Definition 2.9. A *principal component* is a certain linear combination of the original variables.

$$PC_a = u_{1a} \cdot X_1 + u_{2a} \cdot X_2 + \cdots + u_{na} \cdot X_n$$

where $X_1, \ldots, X_n$ are original variables.

We can also call principal components *factorial axes*. Because in the geometric sense, each principal component is equivalent to a new coordinate axis obtained after the original coordinate axis (original variable) is rotated.

Definition 2.10. A *factorial plane* is a plane consisting of two factorial axes (principal components).

It should be noted that PCA can also project the original data point cloud onto the hyperplane, and it is not necessary to project onto the two-dimensional plane.

Definition 2.11. *The most informative projection plane* is the factorial plane that maximizes the projected inertia (Definition 2.8).

Definition 2.12. The matrix obtained after centering (a zero-meaning process) the elements of the original matrix is called *centralized matrix*.

After defining the most basic concepts, we will use some linear algebra methods to find the best dimensionality reduction space, the principal components of the dimensionality reduction space and the dimensionality reduction projection coordinates of the original data points.

Suppose given a triplet $\{X, M, D\}$, where X is a centralized data matrix with dimension $m \times n$ (m individuals and n variables), D is a matrix of individuals weights with dimension $m \times m$ and M is the metric matrix to compare individuals with dimension $n \times n$.

In the case of PCA, the metric matrix M assumes the Euclidean metric and hence $M = \mathrm{Id}_p$.

If the data matrix X is centralized, the angle between two projected variables (the projected variables represented as a vector in factorial plane) matches the correlation between them.

We can deduce this result from *cosine similarity*, which says that

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\| \cdot \|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2}\ \sqrt{\sum_{i=1}^{n}(B_i)^2}}.$$

This result is also used to measure cohesion within clusters in the field of data mining.

Now we need a matrix that can catch relationships and oppositions of data. And then find the best rotation of original axis that make the new coordinate system represents the most information of the original one.

As we mentioned before, the quantity of "information" is measured using the sum of squared deviations or variance. In PCA, we generally use variance to measure how much information there is. There are many ways to measure information, and variance is used in PCA, because we want the projected projection values to be as dispersed as possible. Imagine if the projections of two points overlap each other, then we cannot distinguish the two in the projection space.

And mathematically, the dispersion can be expressed by variance.

If we want to reduce a two-dimensional space to one dimension, then equivalently, we need to find a one-dimensional basis so that all data are transformed into coordinate representations on this basis, and the variance calculated by the following formula:

$$\mathrm{Var}(X_1) = \frac{1}{m}\sum_{i=1}^{m}(X_{1i} - \mu)^2,$$

where X_1 is the first variable. We are considering the case that reduce the dimension from 2 to 1. So we only need one one-dimensional basis PC_1 and X_{1i} are the projected coordinates of original data points.

Because we have already centralized the data, so the value of μ is equal to 0.

For the problem of reducing the above two-dimensional to one-dimensional, it is enough to find the direction that maximizes the variance. But what if we want to reduce a three-dimensional space to two dimensions? Same as before, first we hope to find a direction that maximizes the variance after projection, thus completing the selection of the first direction, and then we choose the second projection direction. If we still simply choose the direction with the largest variance, it is obvious that this direction and the first direction should be "almost coincident". Obviously, such a dimension is useless, so there should be other constraints. Intuitively speaking, let the two fields represent as much original information as

possible, we do not want a (linear) correlation between them, because the correlation means that the two fields are not completely independent, and there must be repeated representations information.

Mathematically, we can express association in terms of covariance between two variables. Because we have completed centralization, $\mu = 0$ and

$$\text{Cov}(X_1, X_2) = \frac{1}{m} \sum_{i=1}^{m} X_{1i} \cdot X_{2i}.$$

Since we need the projections of the two variables to be orthogonal, we need their covariance to be equal to 0.

So far, we have obtained the optimization goal of the dimensionality reduction problem: reduce a set of n dimensional vectors to k dimensions (k is greater than 0, less than n), and the goal is to select k units (the modulus is 1) orthogonal base, so that after the original data is transformed into this set of bases, the covariance between each variable is 0, and the variance of the variable is as large as possible (under the constraint of orthogonality, the largest k variances).

And this means that we have to diagonalize the covariance matrix.

Suppose we only have two variables X_1 and X_2, then we form them into a matrix X ($m \times 2$) by columns:

$$\begin{pmatrix} X_{11} & X_{21} \\ ... & ... \\ X_{1m} & X_{2m} \end{pmatrix}$$

Because we have centered the matrix. So we can directly calculate the covariance matrix of these two variables by multiplying the transpose of the matrix by the matrix and then multiplying by the coefficient $\frac{1}{m}$.

And then we have the covariance matrix

$$\frac{1}{m} X^t \cdot X = \begin{pmatrix} \frac{1}{m}\sum_{i=1}^{m}(X_{1i})^2 & \frac{1}{m}\sum_{i=1}^{m}(X_{1i} \cdot X_{2i}) \\ ... & ... \\ \frac{1}{m}\sum_{i=1}^{m}(X_{1i} \cdot X_{2i}) & \frac{1}{m}\sum_{i=1}^{m}(X_{2i})^2 \end{pmatrix}$$

The last step will be the diagonalization of this covariance matrix. We need to find a matrix P that can convert the covariance matrix into a diagonal matrix. The first k columns of matrix P are the coefficients of the linear combination that give the principal component.

In summary, we have learned how to find the best k principal components, and now we will generalize it into the more general case. We define MX^tDXM to be a covariance matrix, which preserves the covariance between the original variables and the variance of each variable.

According to the content and conditions above, we can deduce the following results using knowledge of linear algebra (we refer to the book of Greub [4] for the linear algebra knowledge involved).

The following propositions are proven using the basic linear algebra.

Proposition 2.13. Rang$(MX^tDXM) = r$, $r = rang(X)$ and also r is the number of positive eigenvalues and $n - r$ null eigenvalues.

Proposition 2.14. Tr$(MX^tDXM) = \sum_{i=1}^{r} \lambda_i$ where λ_i are r non null eigenvalues.

Proposition 2.15. In the case of PCA, we consider the Euclidean distance, so the metric matrix $M = Id$ and then $(MX^tDXM) = (X^tDX)$

If X is centralized and D is diagonal: (X^tDX) is the covariance matrix of X.

If X is standardized and D is diagonal: (X^tDX) is the correlation matrix of X. (We prefer the correlation matrix because big variabilities do not dominate the analysis.)

Proposition 2.16. If we diagonalize the correlation matrix (X^tDX) (meaning that X is standardized and D is diagonal), then we will get r eigenvalues λ_i and sort decreasingly in the diagonal, $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_r$. Moreover, their corresponding eigenvectors $u_i = (u_{i1}, ..., u_{in})$ are orthonormal and contribute a orthonormal base for individuals.

Proposition 2.17. In general, if we diagonalize (MX^tDXM) with M any matrix, then the result is a little bit different. We will still get r eigenvalues λ_i and sort decreasingly in the diagonal as before $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_r$.

But the eigenvectors are no longer orthonormal and $u_{orti} = M^{-1/2}u_i$

Proof. $|u_{orti}|_M = 1$ and we see $u_{orti}^t M u_{orti} = u_i^t M^{-1/2} M M^{-1/2} u_i = 1$;
$u_{orti} M u_{ortj} = 0$ and we see $u_{orti} M u_{ortj} = u_i^t M^{-1/2} M M^{-1/2} u_j = 0$. □

In fact, every eigenvalue λ_i represents the quantity of information (variance) conserved by factor (principal component) i. And the sum of $\lambda \sum_{i=1}^{r} \lambda_i$ is equal to the total inertial of the data matrix X.

Their corresponding eigenvectors $u_i = (u_{i1}, ..., u_{in})$ are the directions of respective principal component PC_i. Geometrically, we can consider them as the rotation of original axis and they formed the new coordinate system that we need.

We still have a problem, that is: how many principal components should we choose, and what criteria do we use to select the first k principal components?

As mentioned before, in principal component analysis, we use *inertia* (variance) to measure how much "information" each principal component holds during the analysis. It is logical to use variance as a measure of information. Because the larger the variance, the easier it is for us to distinguish different projection points, so it is easier to analyze more valuable information.

But so far, PCA does not have an objective criteria to judge the value of k in the first k principal components that need to be retained.

In other words, researchers completely rely on their own subjective needs for the amount of retained data to determine the value of k. If it is necessary to preserve 90% of the variance, then select the top k principal components whose cumulative variance exceeds 90%. And if only 60% of the variance needs to be retained, then the first k principal components whose cumulative variance exceeds 60% are also selected.

According to practical experience, generally speaking, data analysts only need to retain 80% of the variance to obtain better statistical analysis results.

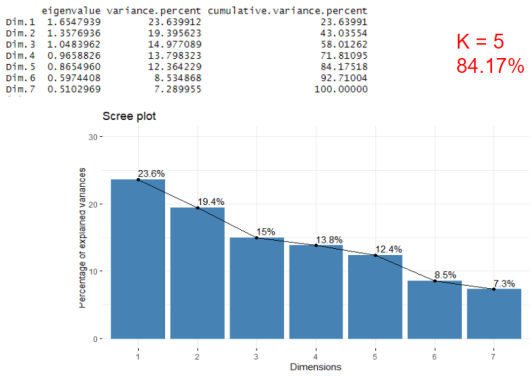

Figure 2.1: Example of principal components ordered in their variances

As this picture showed above, we kept the first 5 ($k = 5$) principal components, thereby preserving 84,17% of the variance of the original data.

However, experience is not always reliable. It may be that for some data sets, only retaining 80% of the variance will cause some key information to be missing. As a result, the analysis results are seriously distorted. Therefore, we need a more objective selection standard that only focuses on the nature of the data set itself, rather than relying entirely on the subjective judgment of researchers.

In Chapter 4 of this work, we will propose a new, more objective judgment method based on the topology of the data set itself. Using this new standard, we can get rid of subjective judgments to choose the principal components we need.

Chapter 3

Topological Data Analysis

3.1 Introduction to TDA

With the recent explosion in the amount, variety, and dimensionality of available data, identifying, extracting, and exploiting their underlying structures has become a crucial problem for data analysis and statistical learning.

Traditional data analysis techniques have not always been able to keep up with the explosion in data volume and complexity because they often rely on oversimplified assumptions. The field of topological data analysis (TDA) attempts to fill this gap by producing a family of techniques derived from the idea that data has a shape that can be rigorously quantified to study data.

TDA constructs simplicial complexes associated with the data and infers qualitative characteristics of the set from the homology of the complexes. These features can quantify complex topological shapes and geometric structures in data to answer questions from the data domain. These data are usually represented as point clouds in Euclidean or more general metric spaces. In this work, we introduce a commonly used topology tool —persistence diagrams. Persistence diagrams represent loops and holes in space by considering the connectivity of data points to obtain continuous values instead of a single fixed value.

TDA is a new field emerging from diverse work in algebraic topology and computational geometry in the 2000s. Although the history of data analysis via geometric methods goes back a long way, TDA really started as a field in topological persistence by Edelsbrunner et al. [1]. This marks the real beginning of TDA. Later Zomorodian and Carlsson's persistent homology [8] made TDA a really powerful technique to use. Although the underlying principles of topological data analysis are not easy to understand, thanks to the existing topological data analysis code files in various programming languages. Even a novice who does not understand can complete the analysis. In this work, the ultra-fast C++ Ripser

package is used as the core computing engine, and the Ripser.py module built in Python is used to implement the analysis work.

3.2 Basic concepts

First, we need to define some basic concepts in algebraic topology, as well as related theorems and conclusions. Because the tools used later are based on these concepts.

We divide these concepts into two parts:

1. Simplicial complexes and filtrations.
2. Homology groups.

According to the definition, in affine space the difference between tow points is a vector, and the addition of a point and a vector yields another point, although addition between points cannot be done.

From this definition, we can deduce the following fact. There are $n+1$ affinely independent points in a k-dimensional Euclidean space ($k \geq n$) if and only if there is no $(n-1)$-dimensional hyperplane that contains $n+1$ points. The hyperplane that contains $n+1$ points needs to be at least n-dimensional.

Definition 3.1. A n-simplex is a n-dimensional polytope which is the smallest convex hull of its $n+1$ vertices. More formally, an n-simplex is determined by a set of points

$$C = \{\theta_0 P_0 + \cdots + \theta_n P_n \mid \sum_{i=0}^{k} \theta_i = 1,\ \theta_i \geq 0,\ i = 0, \ldots, n\}$$

where $\{P_0, \ldots, P_n\}$ are $n+1$ affinely independent points.

Example 3.2. A 0-simplex is a point, a 1-simplex is a line segment, a 2-simplex is a triangle, a 3-simplex is a tetrahedron, and a 4-simplex is a 5-cell.

An n-simplex is the smallest convex hull containing a set of affinely independent points.

Before the definition of simplicial complex, we need to define the faces that have been shared between simplices.

Definition 3.3. The convex hull of any nonempty subset of the $n+1$ points that define an n-simplex is called a *face* of the simplex. Faces are also simplices themselves.

If we wish to study more complex structures, it is not enough to rely on one simplex. We want to study the graphics formed by simplices according to certain rules. And those combinations of simplices ar called simplicial complexes. In mathematics, a simplicial complex is a set composed of points, line segments, triangles, and their n-dimensional analogues.

Definition 3.4. A *simplicial complex* K is a set of simplices that satisfies the following conditions:

1. Every face of a simplex from K is also an element (a simplex) in K.
2. The non-empty intersection of any two simplices $\sigma_1, \sigma_2 \in K$ is a face of both σ_1 and σ_2.

The reason we define simplicial complexes is that we need to create simplicial complexes based on data clouds. Each data point in the data point cloud is usually regarded as a vertex, then we can regard the data point cloud as a set of vertices.

We want to create simplices and simplicial complexes relating data points. The idea is to consider subsets of data points and then find out the possible structures of simplicial complexes.

Thanks to the work of Pavel Aleksandrov, we were able to define the concept of *nerve* for a covering.

Definition 3.5. Let I be a set of indices and C be a family of open subsets $(U_i)_{i\in I}$. The *nerve* of C is a set of finite subsets of index set I. It contains all finite subsets $J \subseteq I$ such that the intersection of the U_i with subindices i in J is non-empty.

$$N(C) := \{J \subseteq I : \bigcap_{j\in J} U_j \neq \varnothing,\ J\ \ finite\ \ set\}.$$

Based on the conception of nerve, we can construct a widely used complex which is called *Čech complex*.

Definition 3.6. Given a finite point cloud X and an $\varepsilon \geq 0$. The *Čech complex* is the nerve of the set of ε-balls centered at points of X.

Čech complex is widely use and captures topological information of a point cloud. But in this work, we will use another complex similar to the Čech complex, which is called *Vietoris-Rips complex*.

The reason of choosing Vietoris-Rips complex is that the Čech complex is more computationally expensive than the Vietoris-Rips complex.

Definition 3.7. If X is a finite point subset in $\mathbb{R}^n$ and given a parameter $\varepsilon \geq 0$, we define the *Vietoris-Rips complex* as

$$VR_\varepsilon(X) := \{\sigma \subseteq X \mid d(x_i, x_j) \leq \varepsilon, for\ \ all\ \ x_i, x_j \in \sigma\}.$$

In order to study the possible changes of a simplicial complex with different values of parameter ε, we need to use a concept in set theory which is *filtration*.

Definition 3.8. Given a simplicial complex C, the *filtration* about C is a family of indexed subcomplexes $F = \{C_i \subseteq C\}_{i \in I}$. F is indexed on an ordered set I such that if $i \leq j$ then $C_i \subseteq C_j$ and also $\varnothing, C \in F$.

There exist $i_0, i_1, \ldots, i_{n-1}, i_n \in I$, $i_0 \leq i_1 \leq \cdots \leq i_{n-1} \leq i_n$ such that

$$\varnothing = C_{i_0} \subseteq C_{i_1} \subseteq \cdots C_{n_1} \subseteq C_n = C.$$

Example 3.9. Given a point cloud $X \subseteq \mathbb{R}^n$, $F_C = \{C_\varepsilon(X) \mid \varepsilon \geq 0\}$ and $F_{VR} = \{VR_\varepsilon(X) \mid \varepsilon \geq 0\}$ are filtrations for the Čech complex and the Vietoris-Rips complex.

Furthermore, we can study the variation of holes in different dimensions in each step. In order to see the appearance and disappearance of holes, we need to understand the following concepts. The following definitions, theorems and propositions are derived from the book of Allen Hatcher [5]. For more information and details, consult this book.

Algebraic topology has two important tools: Homotopy and Homology. We explain a little bit these concepts. When two continuous functions from one topological space to another are called homotopic if one can be "continuously changed" into the other, such a deformation will be called a *homotopy* between these two functions. In practice, there are technical difficulties in using homotopies with certain spaces and their fundamental group is difficult to calculate for higher dimensions. Fortunately there is a more computable alternative than homotopy groups: the homology groups $H_n(X)$.

Before the definition of homology groups, is necessary to know some preliminary concepts such as n-cell, CW-complexes and *boundary functions*.

From the part of cell-complex in [5], we can know that there is a more familiar way to constructing the torus $S^1 \times S^1$ by identifying opposite sides of a square. More generally, an orientable surface M_g of genus g can be constructed from a polygon with $4g$ sides by identifying pairs of edges. A simple way to understand the genus is the number of "holes" of a surface. For example a sphere has genus 0 because of no hole exist, and a torus has genus 1 with the hole seems like the one a donuts has.

It is possible to express a topological surface such like a torus with a polygon. The $4g$ edges of the polygon then become a union of $2g$ circles in the surface that all intersect in a single point. The interior of the polygon can be thought of as an open disk, or a 2-cell, attached to the union of the $2g$ circles. We can also consider the union of the circles which formed a 2-cell as being obtained from

their common point of intersection, by attaching $2g$ open arcs, or 1-cells. And also we can regard the two extremes of an closed arc (closed 1-cell) as two discrete points, or 0-cells.

Until here we get the idea of n-cell and now we consider that n-cells have a frontier (or boundary). A 0-cell is a vertex; an 1-cell is an arc with two 0-cells as boundary; a 2-cell is a surface whose frontier consists of a linear combination of several 1-cells; a 3-cell is a three-dimensional object with a linear combination of 2-cells as boundary.

Attaching n-cells is to link the boundary of one of them with the other. A complex formed by attaching the n-cells for various values of n is called a *cell complex* or *CW-complex*.

Example 3.10. If we attach a 1-cell to a 0-cell, we will obtain a structure that the two extremes of that 1-cell coincide and match in the same vertex (that 0-cell). It seems like a cycle in graph theory.

If we attach a 2-cell to a 0-cell, we obtain a sphere (the surface of a hollow ball) because we linked the boundary of the 2-cell with the vertex 0-cell.

What we introduced above are two simple examples of $CW - complex$.

Definition 3.11. An *n-dimensional hole* (H_n) is a hole formed (or restricted) by the frontier (or boundary) generated by an n-cell. The 0-dimensional holes (H_0) are the connected components, usually a vertex. The 1-dimensional holes (H_1) are the holes formed by linear combinations of 1-cells. The hole exists in a cycle which is formed by the concatenation of edges (1-cells). The 2-dimensional holes (H_2) are the holes formed by linear combinations of 2-cells. The interior of a hollow sphere is the most common example.

We let C_n be a free abelian group with basis of n-cells. Thus a C_0 group is a free abelian group with a basis of 0-cells that are vertices. Then we can define a homomorphism (called boundary function) $\partial_i : C_i \to C_{i-1}$ by sending the basis elements of i-cells to $(i-1)$-cells.

For example, we can define a chain of homomorphism as follows:

$$C_i \xrightarrow{\partial_i} C_{i-1} \xrightarrow{\partial_{i-1}} \cdots \longrightarrow C_2 \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0$$

By the proposition proven in the book of Allen Hatcher [5], we know that $Ker\ \partial_i$ is the frontier of i-dimensional holes (H_i). And also that $Im\ \partial_{i+1}$ is the boundary of H_i filled with no hollow space, so to say H_i is no longer a hole.

Definition 3.12. The *n-dimensional homology group* (H_n) is a quotient group defined as

$$H_n(X) := Ker\,\partial_n / Im\,\partial_{n+1},$$

which is used to find the n-dimensional hole (H_n) (defined in 3.11) in the given topological space X.

We can summarize the key idea of homology groups with the phrase "Find the holes".

3.3 Persistent homology and persistence diagram

In this section, we define the core conception of TDA consulting the article of H. Edelsbrunner [2].

As we have already defined *filtration* in Definition 3.8, we have $\varnothing = C_{i_0} \subseteq C_{i_1} \subseteq \cdots C_{n_1} \subseteq C_n = C$, we apply the homology functor, which for each space gives a vector space and for each inclusion gives a linear map:

$$0 = H(C_0) \to H(C_1) \to \cdots \to H(C_n) = H(C)$$

referring to this sequence as a *persistence module*.

As we defined above in 3.11 H_n is the n-th homology group with n the dimension. We assume coefficients in a field F, so that $H_n = F \bigoplus F \bigoplus \cdots \bigoplus F = F^{\beta_n}$ is a vector space over F, with β_n = rank H_n known as the n-th *Betti number*. (The n-th Betti number refers to the number of n-dimensional holes on a topological surface.)

It is instructive to split the module into *indecomposable summands* of the form

$$0 \to F \to \cdots \to F \to 0.$$

There is a unique such decomposition whose direct sum gives the original module. Each summand can be interpreted as the *birth* of a homology class at its first non-zero term and the *death* of the same class right after its last non-zero term.

In other words, we can define it using mathematical language more precisely.

Definition 3.13. Given a simplicial complex C with $F_C = \{C_i \subseteq C\}_{i \in I}$, every $h \in H_n(C_i)$ is an n-dimensional hole of the homology group in the subspace C_i.

The *birth time* of hole h is the first time j in which h appears as an n-dimensional hole. We define the homomorphism $f_{i,j} : H(C_i) \to H(C_j)$ so that

$$T_{birth}(h) := inf\{j \in I \mid h \in Imf_{j,i}\}.$$

The *death time* of hole h is the first time j in which h disappears and no longer is an n-dimensional hole, so that

$$T_{death}(h) := inf\{j \in I \mid h \notin Imf_{i,j}\}.$$

Hence the *persistent homology* is the homology where the *persistence* of a hole refers to the time between its birth and its death. And we use persistence to assess the variation of holes along the process of the related filtration.

Definition 3.14. A *persistence diagram* is a two-dimensional diagram with the birth time of n-dimensional hole (H_n) in the abscissa axis and the death time of n-dimensional hole (H_n) in the ordinate axis.

The coordinates of points in this diagram are represented by (*birth*, *death*).

Example 3.15. Here we draw a persistence diagram with the principal components we have. In this diagram we have two holes: 0-dimensional holes H_0 and 1-dimensional holes H_1.

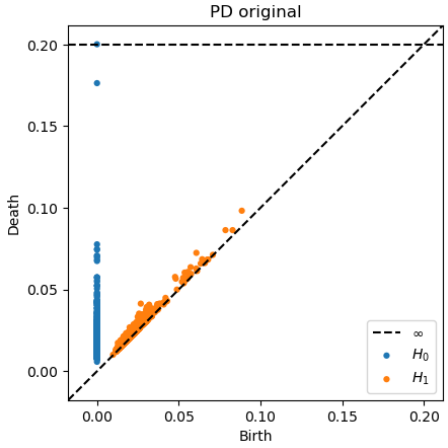

Figure 3.1: Persistence diagram with H0 and H1.

We can observe that all H_0 have the same birth time because the data points exist at the beginning and then they die along the process of filtration. The blue vertical column is formed by them. And the orange points represent the $H1$ holes, which do not follow a certain behavior like H_0.

Definition 3.16. The *i-th post-removal persistence diagram* is the persistence diagram generated by removing all the PCs in the trajectory (see 4.2) until this hierarchy and also the i-th principal component from the rest of the principal components given as the results of previous hierarchies.

We define I as the set of removed PCs for the i-th post-removal persistence diagram and J as the set of removed PCs in previous hierarchies until now (has the same elements with the trajectory until now). So we have:

$$I = \{PC\ i\} \bigcup \{J\}.$$

Example 3.17. Given a set of five principal components named by numbers from 1 to 5, in the first hierarchy of DHA we chose the PC 5 and remember this result (keep it in the trajectory defined in 4.2) and continue the operations of DHA until the end.

In this case the trajectory is $\{5, 3, 2, 1, 4\}$. So that in the first hierarchy, we owned all PCs and the 5 post-removal persistence diagram is the one we look for because it satisfied the criterion established at the beginning. In the second hierarchy, we remembered the previous result so that we had one less and just considered four PCs $\{1, 2, 3, 4\}$. Hence the second one is 3 post-removal persistence diagram generated by the removing PC 5 (already removed from the original data when the second hierarchy began) and PC 3 (selected in this hierarchy for compliance with the criteria established at the beginning). In the third hierarchy we had 2 post-removal diagram generated by the removing of PCs $\{5, 3, 2\}$, and so on.

So that in the i-th post-removal persistence diagram not only we removed PC i but also we removed the previous PCs we have already known.

3.4 Wasserstein distance and bottleneck distance

There are several methods to measure the difference between persistence diagrams. For TDA, we have two common measures widely used: Wasserstein distance and bottleneck distance.

In mathematics, the Wasserstein distance is a distance function defined between probability distributions on a given metric space. This distance is also called "earth mover distance" because we can consider the distribution as a unit amount of earth (soil) piled on M, and the metric is the minimum "cost" of turning one pile into the other, which is assumed to be the amount of earth that needs to be moved times the mean distance it has to be moved.

Definition 3.18. Given a metric space (X, d) that is a Radon space, for $n \in [1, \infty)$, the Wasserstein n-distance between two probability distributions μ and ν on X with finite n-moments is

$$W_n(\mu, \nu) = (inf_{\gamma \in \Gamma(\mu,\nu)} E_{(x,y)\sim_\gamma} d(x, y)^n)^{\frac{1}{n}},$$

where $\Gamma(\mu, \nu)$ is the set of all couplings of μ and ν. A coupling γ is a joint probability measure on $X \times X$ whose marginal distribution are μ and ν on the first and factors respectively.

Their marginal distributions are

$$\int_X \gamma(x, y)dy = \mu(x), \qquad \int_X \gamma(x, y)dx = \nu(y).$$

We use this distance as the metric function between two diagrams because we can match two diagrams and then calculate the minimum cost of transport one into other. The parameter n is used to determine the parameter for moment. Wasserstein distance is a family of distances with different parameters, and the bottleneck distance is a particular case of Wasserstein distance with the parameter equal to ∞.

Bottleneck distance is computationally more costly than Wasserstein distance because to consider the parameter equal to ∞ costs more time than a finite parameter. Consequently, we use Wasserstein distance in this paper with parameter equal to 1 because the default parameter used in Python is 1.

Example 3.19. We show an example of bottleneck distance for a persistence diagram using DHA-bottleneck distance (introduced later in Section 4.2).

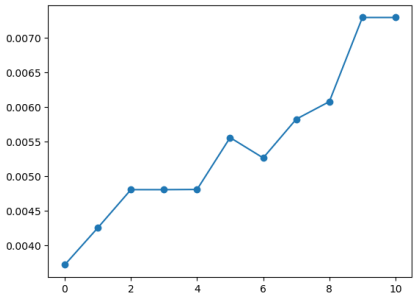

Figure 3.2: Bottleneck distance between the original diagram and a post-removal diagram.

This figure shows the curve which represents the bottleneck distance between the original persistence diagram and the post-removal persistence diagram for each hierarchy (defined in 4.1). The algorithm we used here will be introduced later in Section 4.2.

So we only see here that the distance of a post-removal diagram from the original one is in general monotonously increasing with the increase of PCs removed from all. This result makes sense because it is logical to think that the distance will be larger with more variables (PCs) removed and also the diagram will be more different from the original.

Moreover we will see the same figure with Wasserstein distance in the example 4.9, and that has the curve softer than we have seen here.

3.5 Total persistence

One of the disadvantages of TDA is the difficulty to interpret and explain the meaning of the results we obtained. For example, once we made the persistence diagram, we cannot find out a conclusion directly from this diagram.

So we need a method to quantify the diagram with a certain number that allows us to do a numerical analysis and what we found is a intrinsic topological property called *total persistence*. We used the concept defined in the work of M. Rucco [6].

Definition 3.20. *Total persistence* is a numerical property of the persistence diagram defined by

$$Total\ Persistence\ := \sum_{i \in I} l_i \qquad with\ l_i = death_i - birth_i,$$

where I is the set of hole indices.

So the total persistence is simply the sum of the persistence time of all holes once generated during the process of filtration. It obviously is a real positive number. So we can consider it as a numerical descriptor of a persistence diagram.

3.6 A measure of information: entropy

In information theory, the ntropy of a random variable is the average level of "information" inherent in the variable's possible outcomes. And this average information level is determined according to the unexpectedness and uncertainty of the event. We need to notice that events with a smaller probability will provide more information entropy, because an ordinary event will not make people feel surprised, nor will it make people feel that something special has happened. And once an unusual event occurs, it is natural to notice possible changes, and this provides us with more information.

A very simple example is: when a patient goes to the doctor. When the doctor asks what symptoms he has, if the patient answers: "When I feel hungry, I want to eat food. When I feel thirsty, I want to drink water." Then the doctor cannot get a lot of valuable information from this answer, because the event answered by the patient is a very common and high probability event. Of course, this answer also contains certain information. For example, "want to drink water" means that the patient is not a patient with rabies or other diseases that cannot drink water. But obviously, the amount of information contained in this answer will be very little. But if the patient answers: "When I was hungry, I did not have the appetite to eat. When I was thirsty, I did not want to drink." In this case, the doctor can

pass these rare symptoms to quickly and accurately judge the disease that patient suffers. Because this is not a common occurrence, more information is included to allow doctors to specify several rare and special diseases.

According to this idea, Claude Shannon has defined the information entropy in his paper 'A mathematical theory of communication' [7] as a reasonable measure of information contained. We use his definition here.

Definition 3.21. Given a discrete random variable X with the distribution according to $p : X \to [0, 1]$ (i.e., the probability), the *Information entropy* is defined as

$$H(X) := -\sum_{x \in X} p(x) \log(p(x)) = \mathbb{E}[-\log(p(X))].$$

The base of logarithm can be defined depending on the situation we study.

By the definition of information entropy, we can see that the event with lower probability will give a greater information.

3.7 Persistence entropy

Scholars who use TDA for data analysis are inspired by the definition of information entropy (defined in 3.21), thus defining a new entropy based on total persistence and persistence of holes (defined in 3.13) to measure the amount of topological information. This entropy is called *persistence entropy* and we will use the definition in the work of M. Rucco [6].

In the case of persistent topology, we define the probability of n-hole.

Definition 3.22. The parameter for the distribution is defined as:

$$p_i := \frac{death_i - birth_i}{Total\ persistence}.$$

Before the definition of persistence entropy, we would like to see that this parameter we defined above is a probability.

It is easy to see that:
1) $p_i \geq 0$ because the death time will never be earlier than the birth time.
2) $\sum_{i \in I} p_i = \frac{Total\ persistence}{Total\ persistence} = 1$
3) $P(\bigcup_{i=1}^{n} h_i) = P(h_1 \bigcup \cdots \bigcup h_n) = P(h_1) + \cdots + P(h_n) = \sum_{i=1}^{n} P(h_i)$.

Definition 3.23. *Persistence entropy* is defined as

$$Persistence\ entropy(X) := -\sum_{x \in X} p(x) \log_2(p(x)) = \mathbb{E}[-\log_2(p(X))],$$

where X is a set of holes exist in a certain persistence diagram and $x \in X$ are the holes. The logarithm is with base 2: $\log_2$.

This paper has two core conceptions:

1. Persistence entropy.

2. Directed hierarchical analysis.

Persistence entropy is a useful measure of topological information, and we can define more criteria and tools based on that in the following content of this thesis.

Definition 3.24. An *interval of tolerance* is an interval of real numbers calculated by the mean value of the all differences in absolute value between the original persistence entropy and the persistence entropy of i post-removal diagram in each hierarchy.

$$Interval := [original\ entropy - mean, original\ entropy + mean].$$

Definition 3.25. We define *topological latent dimension* as the last dimension before the difference between the persistence entropy of i post-removal diagram and original persistence entropy varies intensely with the variation larger than the interval of tolerance (in 3.24).

Example 3.26. In order for readers to understand the above two definitions more clearly, here we will give a specific example.

The graphic below shows a blue curve and a red curve. The blue curve represents the persistence entropy generated with DHA-Persistence entropy (we will introduce later in section 4.4) and the red curve represents the original persistence entropy generated with all principal components we have.

Also we need to know that the persistence entropy of a diagram generated by only one principal component (or one variable in the general case) is always null. By the definition 3.11, we know that the 1-dimensional hole $H1$ is formed by the 1-cell and we do not have any $H1$ in the 1-dimensional space. So we always show the total persistence and persistence diagram with $H1$ until where we have two PCs remained.

In this case, we have twelve PCs at the beginning, so that the graphic shows the number of PCs removed from $x = 0$ to $x = 10$.

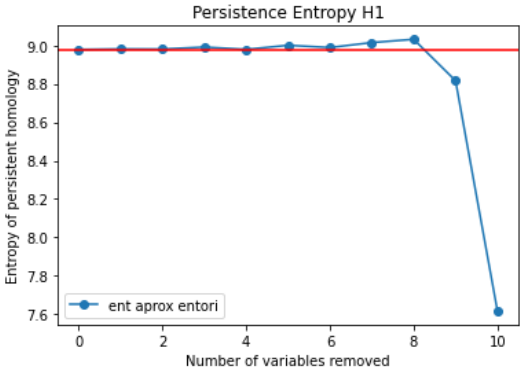

Figure 3.3: A curve of persistence entropy in H1 with DHA-Persistence entropy.

Obviously, in this case, the interval of tolerance (definition 3.24) is [original entropy $+1.06417$, original entropy -1.06417], and the last x with variation contained in the interval is $x = 9$ means we have removed nine PCs and equivalent to remain three PCs. So the topological latent dimension in this case is 9.

Although we found the topological latent dimension based on the topological information using persistence entropy, there is no objective definition to say which is the most proper definition of the topological dimension for a set of data. This is only one possible definition with the point of view on the persistence entropy and my own criterion.

Further more, in the example 3.26 we can observe that seems like where $x = 8$ (equivalent to four PCs remained) also can be considered as a topological latent dimension if we optimize the relation between the count of PCs removed and the their persistence entropy. But for more general cases, we need a objective criterion like *interval of tolerance* (defined in 3.24) to deal with the other situations.

Chapter 4

Directed Hierarchical Analysis

In this chapter, we will introduce a new analysis method that is defined for the first time which is called *Directed Hierarchical Analysis* (DHA). The purpose of this paper is to find a data dimensionality reduction method based on topological persistent homology, persistent entropy and principal component analysis.

First of all, we need to apply PCA over the data base in order to concentrate the information and remove the multicollinearity between the variables. Once we have the principal components, we need to find out how many principal components we need to keep optimally after pooling information using PCA, and learn which ones they are.

As we said before, the traditional way to select principal components depend on the accumulation of variance. More variance always means more information kept in the projection space. We also pointed out that this method depend on the amount of variance that we expected to use for the analysis. So there is not an objective criteria to decide the number of reserved principal components, and it will be determined according to the researcher's needs.

Inspiration from the *jenga* game tells us that we can remove some principal components and then observe the possible variation of topological structure caused by the removing. There will be a spoiled structure or remain stable and keep almost the same structure as before.

One of the innovation in this paper is find out a method that decide how many PCs to keep by looking at the topology itself in the data structure. That is to say, this is an objective standard. The other innovation is the creation of the new analysis method called Directed hierarchical analysis (DHA).

DHA is based on the topological properties and information. In the section 3.3, we have introduced how to capture the topological structure of the data through the birth and death of n-dimensional holes over time (In general they are zero and one dimensional holes).

Definition 4.1. *Directed Hierarchy Analysis* (DHA) is an analysis method that remove one principal component from the rest of them in each step of analysis and then compare the changes occurred or different value of some specific data structure properties properties between the original topological data structure (without removing any principal component) and the post-removal topological structure.

Process of algorithm:

1. Decide a certain criterion and a specific topological structure property for choosing the principal component.
2. At the beginning, there are n principal components generated by PCA.
3. Remove the first principal component and calculate the value of property defined at first.
4. Repeat the third step until we have calculated all values of the property for each removing.
5. Find out which is the value of property that satisfies the criteria defined at first.
6. Discover the principal component removed that corresponds to the generation of the value selected in the fifth step.
7. Remove the principal component discovered in the sixth step and back forward to the second step with $n - 1$ principal components. We refer to this set of steps, steps 2 through 7, as an *hierarchy*.
8. Continue this process until the last one principal component is remained.

We named this method as DHA because of two characteristics possessed: Directionality and Hierarchical (also known as memory or heredity).

Directed means there is an orientation that given by a certain criterion (or rule) that help us to decide which principal component is the one that need to be removed from all. The direction make sure that we will not analyze randomly and always study all of them.

Hierarchical means that each hierarchy (defined in the step 7 of definition 4.1) of the analysis is based on the results of the previous hierarchy. We can continue go deeper into the next hierarchy based on the results of all previous hierarchies.

Definition 4.2. In each hierarchy, we keep the name (or the number) of the selected PC and the order of this hierarchy. Add this PC into a sequence. Continue this operation and every time add the PC selected following the order of correspondent hierarchy until the last hierarchy.

The sequence created by the names of PCs in their order we call that *trajectory* of DHA.

In other words, the *trajectory* of DHA is:

$$trajectory = \{N_i\}_{i=1,2,\ldots,n-1} \bigcup \{N_n\}$$

The N_n is the last PC remained when the $n-1$-th hierarchy has been done.

Example 4.3. Given 7 principal components by PCA. We define the Wasserstein distance between the original persistence diagram and the persistence diagram generated after the removing of one PC as the topological structure property in this case. And the criterion of selection is to find out the minimum Wasserstein distance between two diagrams.

And the first principal component removed by DHA in the first hierarchy is PC 5, then we continue the DHA with PC $1, 2, 3, 6, 7$. At the end of all, we got the trajectory (4.2) of this case which is {5, 3, 6, 1, 2, 4} means in the first hierarchy we removed PC 5 with {1,2,3,4,6,7} remained, in the second hierarchy, base on the removal of PC 5, we removed PC 3 with {1,2,4,6,7} remained. In the third hierarchy, base on the removal of PC 5 and PC 3, we removed PC 6 with {1,2,4,7} remained and so on. Now we can observe the curve of Wasserstein distance generated by removing the PCs in order and their correspondent Wasserstein distance to original diagram in each hierarchy.

If we do not use DHA, we will not have the criterion to detect the principal component wanted and probably we have to use all of the values of certain topological structure property. And then, as we did not choose any principal component of them, we should remove two of them from the original n PCs. Also the Non-Hierarchical makes that the result we got before will not help us to get further more for the following steps. Iterating over all possible results at each step makes the analysis very inefficient. Further more, it will let the values that are not related to my research purpose interfere with the final analysis results, resulting in distortion of the results.

We have told that DHA needs a topological structure property and also a certain criterion, in this paper, we will use the following DHA-based methods with different criteria and properties.

Based on DHA: minimum wasserstein distance, minimum bottleneck distance, maximum total persistence, maximum persistent entropy and the combination of these methods. These methods are based on the concepts introduced in the previous chapters.

We will describe these DHA-based methods in detail in each section of this chapter.

4.1 Distance centroid method

As we introduced in the section 3.4, Wasserstein distance and Bottleneck distance can used as the metric function that measure the difference between two persistence diagrams.

To find out the latent dimension by selecting the principal components, we can compare the difference between the original persistence diagram and the post-removal diagram.

Here we define the simplest method to do the comparison: *Distance centroid method*.

Definition 4.4. *Distance centroid method* is an analysis method that in the step i ($i \in \{1, 2, \ldots, n-1\}$). We remove i principal components from all of them and then calculate the distance (bottleneck, Wasserstein, etc) between the i post-removal persistence diagram (definition 3.16) and the original persistence diagram. We consider all possible cases where i principal components are removed and calculate the mean value of these distances between the persistence diagrams. We call this mean of distance as *centroid*.

Once we have all distance centroid in each step, we can draw a curve that represents the mean distance between the i post-removal diagram and the original one.

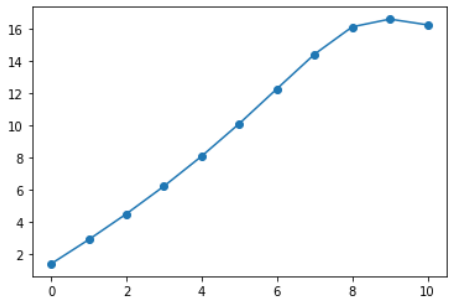

Figure 4.1: A curve of distance centroids

Example 4.5. Here is an example of the result of distance centroid method. The blue curve represents the mean of Wasserstein distance between the i post-removal diagram and the original one with twelve principal components. The abscissa axis x represents the number of principal components removed and where $x = i$ means that we take out $i + 1$ principal components. So that when $x = 0$, we calculate all possible twelve cases of remove just one principal component from the all.

We can observe that this curve rises almost following the same slope before $x = 8$ (the distance centroid with nine PCs removed), and then the slope became

smaller and reached the maximum value in $x = 9$ (ten PCs removed).

According to the hypothesis, the distance centroid will keep going up and the curve should never go down. Because whenever more PCs is removed, the resulting persistence diagram should be farther away from the original image. So we can assume from this phenomenon that maybe something was happened when the curve reached the highest point. And that dimension is what we really interested in.

Proposition 4.6. *Distance centroid method* is extremely inefficient and we will prove this fact.

If we have n principal components after the PCA. And we would like to find out the number j when we removed j PCs, we can reach the dimension in which the Wasserstein distance between j post-removal diagram and the original one does not behave like previous steps.

With Distance centroid method, in the first step, we need to remove 1 PC from n and then calculate the Wasserstein distance between the 1 post-removal persistence diagram and the original. So that in the first step we need to choose one different PC n times and then calculate the distance centroid.

In the second step, we have to remove 2 PC and if we want to iterate over all possibilities, we need to calculate $\binom{n}{2} = \frac{n!}{2!(n-2)!}$ times and then obtain the distance centroid.

In the general case with i PCs removed, the computation times will be $\binom{n}{i} = \frac{n!}{2!(n-2)!}$.

To complete the analysis using this method, the computation will be a waste of time because it is necessary to iterate all possibilities and the time is:

$$TIME = \sum_{i=1}^{n-1} \binom{n}{i}$$

The computational complexity is factorial which is the most complex type.

The inefficiency of distance centroid method prompted us to think and use a new method to reduce computing time and improve efficiency. That is DHA.

4.2 DHA with Wasserstein distance

In fact, their is no necessity to iterate all possible cases if we want to find the dimension in which the behaviour of distance (Wasserstein, Bottleneck, etc) has changed and behaves differently than the previous steps. For example, in the first time of removing PC. We can only consider among the principal components,the

one whose distance between the 1 post-removal diagram and original diagram is closest. Assume that PC is the fifth of n. And in the second step, we remember that in the first hierarchy, we have already known the PC 5 and 1 post-removal diagram with PC 5 removed is the closest one to the original diagram. So we do not have to consider the combination consist of PC 5 with other PCs and PC 5 can be removed from the rest. So that in the second hierarchy, we only need to consider $n-1$ PCs and do the same as the first hierarchy.

Definition 4.7. *DHA-Wasserstein distance* is a variant method based on DHA with Wasserstein distance between the post-removal persistence diagram and the original persistence diagram as the topological structure property and the criterion is always choose the PC which causes the smallest distance to the original diagram after the removing.

Topological property: Wasserstein distance between the original and post-removal

Criterion: Choose the PC i such that the Wasserstein distance between i post-removal diagram and the original is smallest.

Proposition 4.8. *DHA* is more efficient than distance centroid method. The time we need to complete the analysis is:

$$TIME = n - 1$$

The computational complexity is linear so it is efficient.

This *TIME* not only for just one variant method of DHA but also for the whole family of methods based on DHA.

We have mentioned an example 4.5 to explain How does it work work.

Example 4.9. Here we can do a comparison between the results of the same subject with distance centroid method and DHA-Wasserstein distance method.

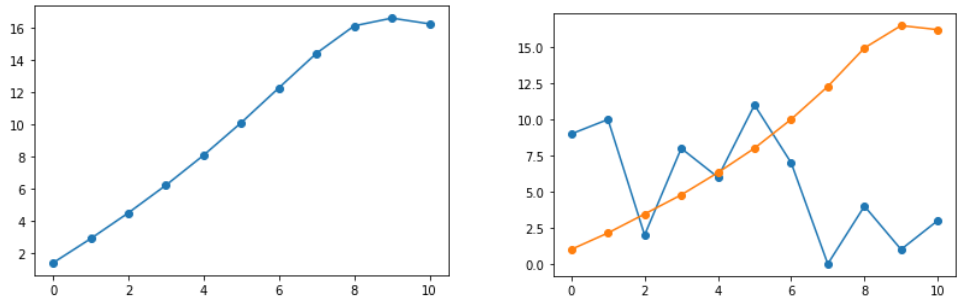

Figure 4.2: Comparison between two methods using the same data with H0.

Both methods use the same data from a subject, and also use the persistence diagram of 0-dimensional holes $H0$ (in section 3.2). The right figure has two curves, the orange one represents the distance centroids in each step and the blue one represents the trajectory (see 4.2), means that the order of removing principal components with DHA-Wasserstein distance. In the first hierarchy, we removed the PC 9, in the second we removed the PC 10 and the final one is PC 3.

We can see that the DHA - Wasserstein distance has generated almost the same the curve as the left one.

The criterion of DHA-Wasserstein distance is choosing the smallest distance not the biggest distance because we want to find the PC which matter less and unrelated with the topological structure. We do our best to make sure the structure be stable as before, not to destroy it with the most extreme PC.

The reason for choosing the Wasserstein distance instead of the bottleneck distance is as stated in 3.4, because bottleneck is a special case of Wasserstein, which is the Wasserstein distance when the parameter value is equal to positive infinity. Wasserstein distance is more proper for the generality.

In this paper, we use the Wasserstein distance with parameter equal to 1. Because of the default parameter of Wasserstein distance function defined in Python is 1.

4.3 DHA with total persistence

We have already introduced the concept of total persistence in the section 3.5. As we also told in Section 3.7, these two conceptions are related deeply.

Persistence entropy contains the topological information of a topological structure, total persistence is used to calculate that entropy. And the persistence entropy only depends on the persistence of each n-dimensional hole and total persistence. So that persistence also contains some parts of topological information. From here, there is the idea of the new method.

Definition 4.10. *DHA-Total persistence* is a variant analysis method based on DHA with total persistence of each i post-removal diagram as the topological structure property and the criterion is always choose the PC which causes the maximum total persistence.

Topological property: Total persistence of each i post-removal diagram.

Criterion: Choose the PC i such that the i post-removal diagram has the maximum total persistence.

Attention, in the last section, we used the criterion which select the PC with minimum Wasserstein distance but in this case we changed ourselves to choose

the maximum value of total entropy because we want to keep as much as possible the topological information contained by total persistence.

Example 4.11. There are two graphics that show two results of two different subjects with DHA-Total persistence.

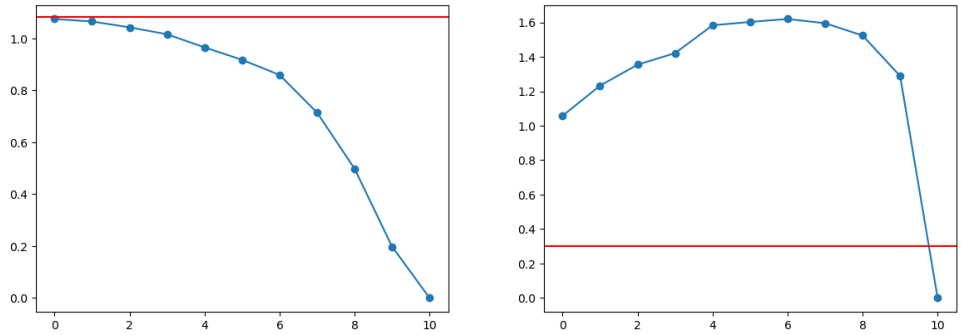

Figure 4.3: Comparison between two subjects with DHA-Total persistence with H1.

The left one is the result generated with the *subject* 30 and the right one is the result generated with *subject* 26. Have to be aware of that the *subject* 26 is an outlier among all observations of our data base.

The red horizontal line represents the total persistence of the original persistence diagram. And both blue curves are formed by the total persistence of i post-removal persistence diagram found by DHA - Total persistence.

We can find observe the blue curve in the left figure represents a monotone decreasing from the original total persistence. on the other hand, we can see this curve behaves strangely because the total persistence found in the first hierarchy has jumped to a extreme high altitude from the original. And in almost every hierarchy, the total persistence we have chosen are above the red line until the giant descent and then went down the red line.

4.4 DHA with persistence entropy

In this section we go a step further and define the final variant method (used in this paper) based on DHA with the same reason as the DHA - Total persistence.

Definition 4.12. *DHA-Persistence entropy* is a variant analysis method based on DHA with persistence entropy as the topological structure property and two criteria choosing the maximum value of persistence entropy or the value of persistence entropy which is the closest to the original entropy.

Topological property: Persistence entropy of i post-removal persistence diagram.

Criteria:

1. Choose the PC i such that the i post-removal diagram has the maximum persistence entropy.

2. Choose the PC i such that the i post-removal diagram has the persistence entropy which is the closest to the original entropy.

The first criterion based on the idea of DHA-Total persistence which keep the topologically most informative choice. And serves for the detection of the outliers and atypical observation in the data base because we always select the extreme large value of persistent entropy in each hierarchy. It can also be used to find out the dimension that the topological structure of data reach the most informative state which can help us to apply some topological analysis about that, like DHA-Total persistence.

The second criterion is used to find the topological latent dimension (defined in 3.25) that always choose the PC that produces the less variation from the original in persistence entropy.

Example 4.13. Here we only show an example for the DHA-Maximum persistence entropy which means we establish the first criterion. For the example of DHA-Approximate original entropy with the second criterion, see example 3.26.

In this example, we put two graphics of the DHA-Maximum persistence entropy with the same data.

Both graphics have one blue curve which represents the persistence entropies of each i post-removal diagrams, the red horizontal line which represents the original persistence entropy. And the abscissa axis represents the number menus 1 of the quantity of PCs removed. In other words, $x = j$ means $j + 1$ PCs removed from all.

The left figure shows the number of removed PCs from 1 to 11, but as we said in the example 3.26, we actually need just from 1 to 10 because when eleven PCs have been removed, only one PC is remained and will never form $1 - cell$ or 1-dimensional hole. So that the total persistence or the persistence entropy with $H1$ and one PC left will always equal to zero.

For the reason of visibility, we put the right figure without the last hierarchy.

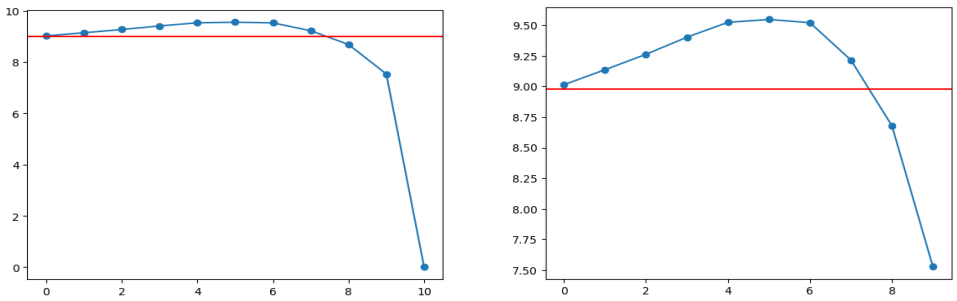

Figure 4.4: DHA-Maximum persistence entropy with H1.

From the right figure, the increasing of persistence entropy can be obviously observed and get the maximum value where $x = 5$ (means six PCs have been removed and equivalent to other six PCs left) and then start decreasing monotonously. So we can conclude that we only need six specifics PCs which are not be removed to find out the possible maximum topological information (see the section 3.6). And these 6 PCs remained are the principal components who concentrate the most topological information.

We say that this highest point of persistence entropy as a possible maximum entropy because we are unable to decide that there is no another order of removing the PCs can get larger entropy. The only thing we can say is the highest point we found probably is the maximum with DHA-Maximum persistence entropy. And we will mention this disadvantages of DHA in the conclusion.

The trajectory of this case is {0, 3, 2, 1, 5, 4, 8, 10, 9, 7, 6, 11}, so that we know the first six PCs {0, 3, 2, 1, 5, 4} do not contain so much topological information and the last six {8, 10, 9, 7, 6, 11} are the most informative.

So we succeed to find the principal components given by PCA which are the most topologically informative. And this method contributes the theoretical and practical basis to discover the order of topological information instead of the order of variance information used in traditional PCA.

4.5 Combined DHA methods

In the final section of this chapter, we create a method that combining the methods mentioned above.

In the section 3.6, we provided a method to measure the topological information using persistence entropy. And the main objective of this paper is data dimensionality reduction with latent dimension. As we defined the topological latent dimension in 3.25, the reduction and the latent dimension are based on persistence entropy, in other words, topological information contained.

So we consider the DHA method combined with Wasserstein distance, total persistence, maximum persistence entropy, approximate original entropy all based on the persistence entropy as the topological structure property.

Definition 4.14. *Combined DHA method based on persistence entropy* is a variant method based on DHA.

In the first step we calculate the trajectories (definition 4.2) respectively produced by DHA-Wasserstein distance, total persistence, maximum persistence entropy, approximate original.

And then we calculate the persistence entropies of post-removal diagrams according to these trajectories.

Example 4.15. Here we put two graphics of the combined DHA method based on persistence entropy to make a clear and understandable explanation.

The left figure uses the data of *subject* 25 and the right one uses the data of *subject* 26 which is an outlier among all observations in our data base. The abscissa axis represents the number of PCs removed. Where $x = j$, we removed j PCs. And we consider the case of $H1$ so there is no necessity to express the $x = 11$ because it is impossible to create $H1$ with one PC (variable/dimension) remained, so the persistence is always null.

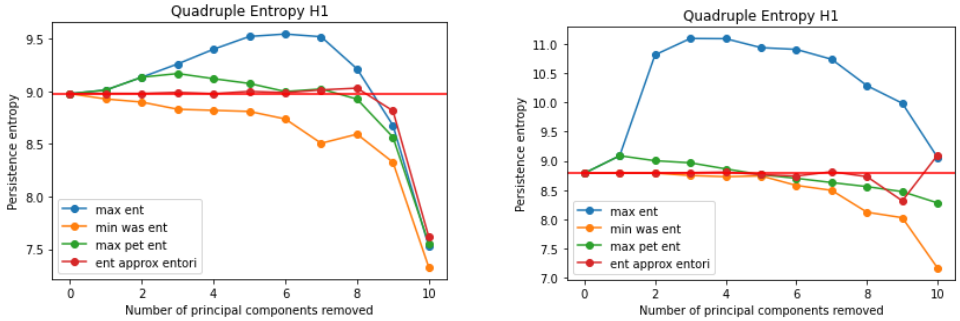

Figure 4.5: Quadruple Entropy for subjects 25 and 26 with H1.

First of all, we explain the meaning of these graphics. For the left graphic, the blue curve represents the persistence entropy calculated using the trajectory of DHA-Maximum persistence entropy. The orange curve represents also the persistence entropy but calculated using the trajectory of DHA-Wasserstein distance ({9, 10, 7, 8, 2, 6, 11, 0, 1, 3, 4, 5}). We know that DHA-Wasserstein distance is used to find the minimum Wasserstein distance between the original and post-removal diagram. But in this case, we do not calculate the distance, but the persistence entropies according to the trajectory found. And the green curve made by the

trajectory of DHA-Total persistence, the red curve drew by the trajectory of DHA-Approximate original entropy ({6, 0, 9, 4, 5, 11, 1, 3, 2, 8, 7}).

Obviously,it can be observed that the blue curve established a supreme limit and other curves are all above that. In most circumstances, the blue curve restrict the others because it was generated by the trajectory of DHA-Maximum persistence entropy ({0, 3, 8, 7, 10, 9, 4, 5, 11, 6, 1, 2}). And the green curve is located in the second highest position because in every hierarchy of DHA-Total persistence we select the PC which made the maximum total persistence (trajectory={0, 3, 8, 7, 10, 9, 4, 5, 11, 6, 1}). So it is reasonable that two curves blue and green are the highest curves.

The red curve approximates a lot the red horizontal line (represents the original entropy) because in every hierarchy we choose the PC with less variation from the original persistence entropy, so it ought to be the most similar curve to the horizontal red line.

And the orange curve behaves a monotonously decreasing from the first hierarchy because we can see that the graphic of DHA-Wasserstein distance of example 4.9 behaves almost the same monotonously decreasing curve. It make sense because in each hierarchy of DHA-Wasserstein distance, the Wasserstein distance is farther and farther away from the original persistence diagram. So that is the reason of the behavior of orange curve like that.

For the right figure, as we already known that this subject is an outlier, such these strange curves do not surprise us. We can see that the green, red, orange curves behave the same as the left image. But the blue curve is extremely high compare with the others. And from here, we can deduce that *subject* 26 has extreme high values of persistence entropies in each hierarchy. And the *subject* 25 does not have those.

So this example also explain the fundamental reason of the outliers detection using the extreme high values of persistence entropies.

Chapter 5

Application of DHA in a specific dataset

5.1 Origin of dataset

The dataset used in this study was a preprocessed version of a dataset collected during a series of experiments carried out by Dr. Ignasi Cos at the Center for Brain and Cognition of Universitat Pompeu Fabra. The experiments were approved by the Clinical Research Ethics Committee (CEIC-Parc Salut Mar) of Universitat Pompeu Fabra-Hospital del Mar with reference number 2015/6085/I, and the methodology was designed in accordance with the corresponding directives and regulations.

The original neuroscience study was described in the following work: Cos I, Deco G, Gilson M (2021): Behavioural and neural correlates of social pressure during decision-making of precision reaches. DOI:10.21203/rs.3.rs-1974463/v1, and an exploitation of the dataset with methods from topological data analysis was carried out in FerrÃ A, Cecchini G, Nobbe Fisas FP, Casacuberta C, Cos I (2023): A topological classifier to characterize brain states: When shape matters more than variance, arXiv:2303.04231 [cs.LG].

5.2 Research objectives

The study by Cos et al. (2021) was carried out with eleven participants. High-density electro-encephalograms (EEG) were recorded during 1200 ms from human participants during a decision-making task in which motivation was modulated via social pressure. The manipulation of motivation was performed by means of a function of the participant's aiming accuracy with respect to that of a virtual

partner. The purpose of simulated partners was to introduce an implicit bias to modulate the participant's motivation to reach more accurately.

Each participant performed two sessions of six blocks each, with each block consisting of 108 trials. The six blocks were distributed into two groups of three. Each group consisted of one block playing alone and two blocks each alongside a partner of a lesser/higher aiming skill. The goal of this manipulation was to induce three distinct motivational states as a function of the level of social pressure exerted. Hence, the dataset consists, per trial, of EEGs from a variable number of channels (normally 60 electrodes per participant) lasting 1200 ms each. Each participant performed 12 blocks of 108 trials.

The purpose of the classifier developed in Ferrà et al. (2023) was to ascertain that TDA is a suitable predictor of brain motivational states when applied to the study dataset.

5.3 Quartiles and box plot

First of all, we need to find out the outliers that exist in our data set.

As we said in the section 4.4 and example 4.15. DHA-Maximum persistence entropy serves to detect the outliers based on the selection of most extreme high value of entropy.

Box plot and quantiles can be used as an powerful tool to see directly the distribution of the entropies produced by DHA-Maximum persistence entropy with different subjects. Box plot is a statistical graph used to show the dispersion of a set of data. It can display the maximum (top bar), minimum (bottom bar), median (orange line), and upper and lower quartiles (top and bottom of the box) of a set of data. The first quartile $Q1$ is equal to the number that larger than the first 25% of all the values in the sample arranged from small to large. The second quartile, also known as the median, is equal to the number that larger than the first 50% of all the values in the sample arranged from small to large. The third quartile $Q3$, also known as the larger quartile, is equal to the number that larger than the first 75% of all the values in the sample arranged from small to large. Box plot can be considered as an expression of the distribution of probability density.

In this section, we will show two images of box plots for all subjects with both the H_1 and H_0. As we said in the section 3.3, the persistence diagrams can be made with the birth and death of H_1 or H_0. The most examples we showed above (specially in the chapter 4) are generated with H_1 because their can almost always happen that some H_0 holes with death time equal to infinity, it is equivalent to say that exist some data points are connected components independent of others and never connect with other connected components. And the calculation of total

persistence does not admit the infinity death time. So we need to remove those H_0 holes that will never die. And this can probably cause the lost of information, That is why we use H_1 holes with more frequencies. Here we give two graphics one above the other with H_1 and H_0 respectively.

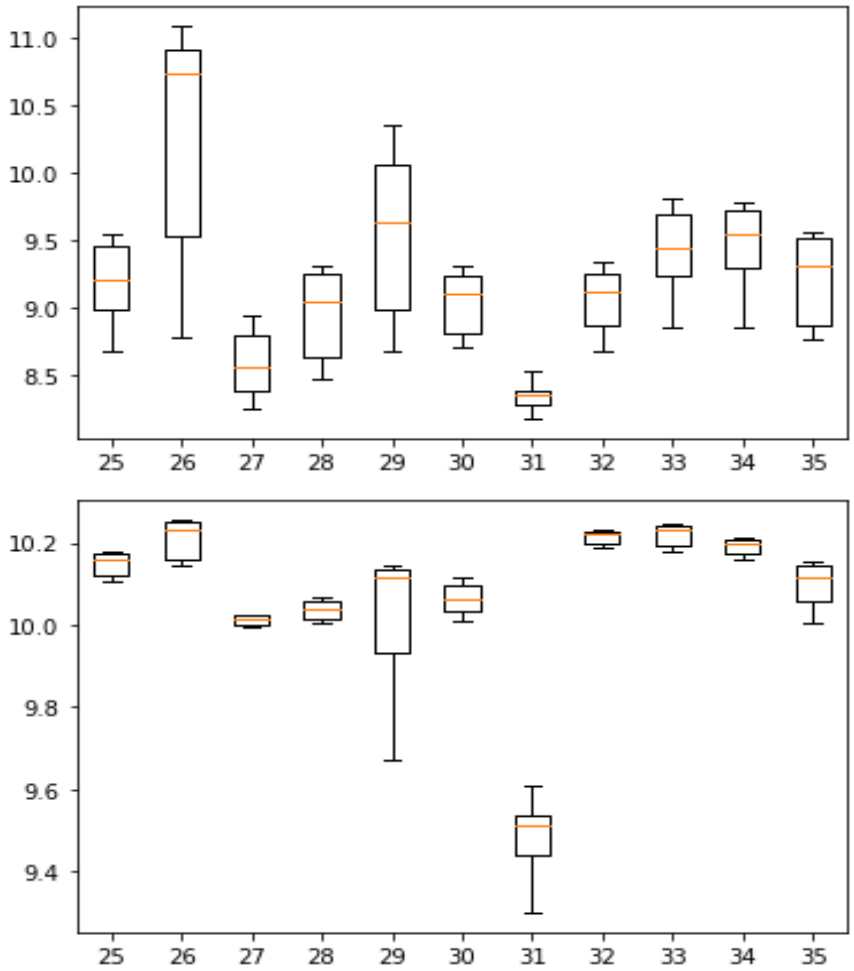

Figure 5.1: Box plots of persistence entropy for all subject with H1 above and H0 below.

First of all, we noticed that although the scale and sizes of these two pictures are different, the interrelationships between subjects are still well preserved. This means that subjects that exhibit anomalous properties in the upper image will still do the same in the lower image.

As mentioned in the previous sections, *subject* 26 is a very typical outlier. It is easy to observe that in the box plot using H1, the distribution of persistence

entropy of *subject* 26 is significantly more disperse than other subjects. The size of the area in which 50% of the data is reserved in the middle of the box plot is jointly determined by $Q3$ and $Q1$. We noticed that $Q1$ was close to 9.5, while $Q3$ was already close to 10.8. The difference between $Q3$ and $Q1$ is even much larger than the difference between the maximum and minimum values of some other box plots. In particularly, the median of *subject* 26 obviously situated at a much higher position than others. There is no other subject has the median more than 10. The *subject* 29 has the same behavior as the *subject* 26 with a widely disperse box and a median significantly higher than others. And we also find the *subject* 31 is the other type of outliers. The box plot is extremely compressed into an area smaller than the most of boxes of other subjects and the median almost coincide with the $Q3$ if we do not enlarge the size of the graphic. From the figure above using H_1, it is clearly to see the subjects {25, 28, 30, 32, 33, 34 ,35} formed the main part of the observations, these observations compare the same behaviors between them. And we consider here the *subject* 27 not also as an outlier because the expansion of box is moderate and is relatively more closer than subjects 26, 29, 31.

For subjects like subject 27 that cannot be judged only by the information in H_1, we can use the results in H_0 for comparative testing. In the second figure, we see the subjects $29, 31$ remain the anomaly and specially the *subject* 31. The abnormality of *subject* 31 in H_0 is even more obvious than the most obvious *subject* 26 in H_1. According to the expansion of the box and the position of median in H_0, we can divide the rest of subjects into two classes consist of normal observations. They are subjects {25, 26, 32, 33, 34, 35} and {27, 28, 30}. As we have already determined the normal observations in H_1, we can find out the intersection between two figures. The criterion is simple and reasonable that we only consider one observation is normal if and only if it has been viewed as normal in both of two figures. So that the normal subjects are {25, 27, 28, 30, 32, 33, 34, 35}. And the outliers are subjects {26, 29, 31}.

5.4 Results of DHA-Wasserstein distance

Since total persistence is a topological property in the persistence diagram, we use it to study how the topology of the data may change as PCs are removed.

In this section, we only show the results of DHA-Wasserstein distance with H_1 because of the reason mentioned above in the section 5.3.

With the division of data set into the normal and outlier classes, we just need to show the figures more representative of each class. For more results, please consult the attached appendix.

Here we show three results of the normal class, there are subjects {25, 33, 35}.

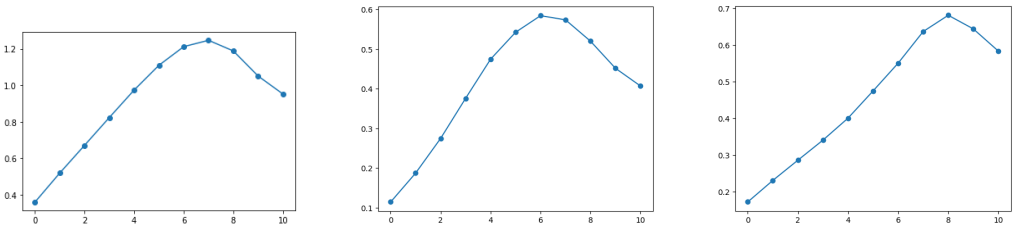

Figure 5.2: DHA-Wasserstein distance in h1 applied on subjects 25, 33, 35.

Every figure has a blue curve which represents the variation of Wasserstein distance between the original persistence diagram and the post-removal diagram (definition 3.16) following the own trajectory (definition 4.2). Where $x = i$ means we removed $i + 1$ PCs.

We can see that all these three figures show a smoothly monotonously increasing with a stable and after reaching the peak, start monotonically decreasing until the elimination of eleven PCs at $x = 10$. The peaks for them are $x = 7$, $x = 6$ and $x = 8$ respectively.

We use DHA-Wasserstein distance analysis because we designed a process and raised a question: If we remove the PC with the least impact (measured by the Wasserstein distance to original diagram) on the overall at each step, will there be a certain node so that when we get there, even the PC with the least impact on the overall. That PC will still have a significantly higher impact on the population than removing the PC in the previous step. If that node does not exist, then the Wasserstein distance between the original diagram and the original diagram should continue to increase monotonically and the increase rate is almost the same, and there should be no intense sudden growth. Generally speaking, as we remove more principal components, the topological structure constructed by the remaining principal components should be more different from the original structure, that is to say, as we remove more principal components , the distance between the post removal diagram and the original diagram should be getting farther and farther. But the facts do not match our conjectures. Because when we reached these three peaks, we found that as we continued to remove the principal components, the distance between the post removal diagram and the original diagram became abnormally closer and closer. In fact, the topology of our remaining principal components cannot be more and more similar to the original one. But this abnormal phenomenon appeared. This means that when the topological structures of the three subjects reach the corresponding peaks, their topological structures collapse, and the calculation of the Wasserstein distance between them and the original cannot correctly and truly reflect the difference between them.

Even this phenomenon also happens with the subjects of outlier class.

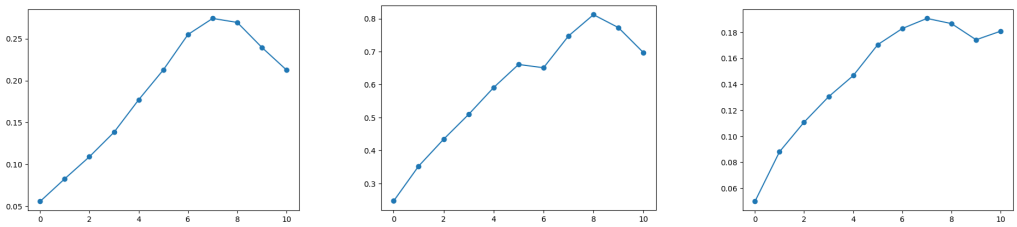

Figure 5.3: DHA-Wasserstein distance in H1 applied on subjects 26, 29, 31.

We can see that in this case, even the curve of *subject* 26 is the same as the previous three pictures, monotonically increasing to the peak, and then looking monotonically decreasing. Even the x value of its peak is equal to 7 (with 8 PCs removed) just like *subject* 25.

Subjects 29 and 30 are outliers, so their curves do not fully comply with the previous rules. The curve of subjects 29 did not satisfy the monotonically increasing property at $x = 6$ before reaching the peak. And *ubjects* 31 did not satisfy the monotonous decrease after peak when $x = 9$. This proves that these two subjects are outliers from another aspect.

Although we obtained these results and know some rupture or structural change happened, we still have no idea about what actually happened when the curves of variation reached their own peak. To find out the reason, we need the intervention of topological information.

5.5 Results of DHA-Total persistence

The purpose of DHA-Total persistence is to find the trajectory of PCs removed that keep the maximum persistence entropy in each hierarchy as we defined in the section 4.3. In this section, we only show the results of DHA-Total persistence with H_1 (the persistence of H_1 holes in diagrams) because of the reason mentioned above in the section 5.3.

Same as above, we put three results of the normal class, there are subjects 28, 30, 32.

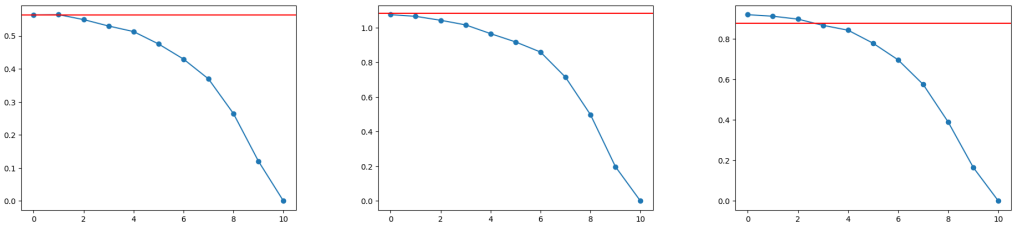

Figure 5.4: DHA-Total persistence in H1 applied on subjects 28, 30, 32.

Every figure has a blue curve which represents the variation of total persistence of the post-removal diagram following the own trajectory (definition 4.2) and the red horizontal line is the total persistence of the original persistence diagram . Where $x = i$ means we removed $i + 1$ PCs.

As these three subjects belong to the normal class, they have almost the same curve performance with respect to their own red line. They always start near the original total persistence (it can be a point above or below), and then remove PCs in order according to their respective trajectories. And their total persistence is a very stable, smooth (meaning there is no intense sudden slope change) monotonous decreasing curve.

We can also observe three more total persistence curves of the subjects in the normal class. Because these three subjects are still normal, but their curve forms have different performances from the above three pictures.

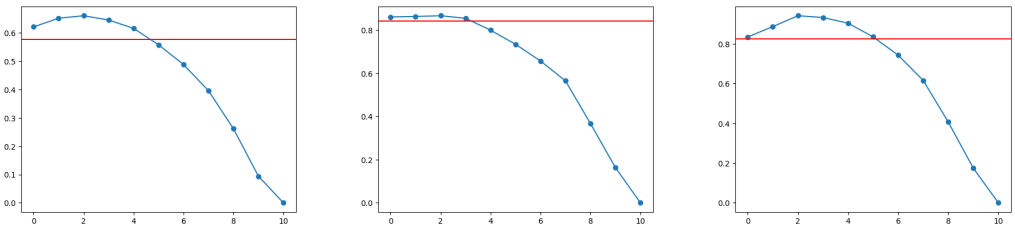

Figure 5.5: DHA-Total persistence in H1 applied on subjects 33, 34, 35.

If we only observe the part of curve below the red horizontal line these six graphics have the same trend and performance. The most significant difference is in the beginning of these curves. The curves began around the original total persistence, But then they start growing briefly until they reach a peak (not the same one as in DHA-Wasserstein distance). Although these three curves seem to show some abnormality, if we compare them with the pictures generated by outliers below, we will find that subjects 33, 34, 35 still belong to the normal class with subjects 28, 30, 32.

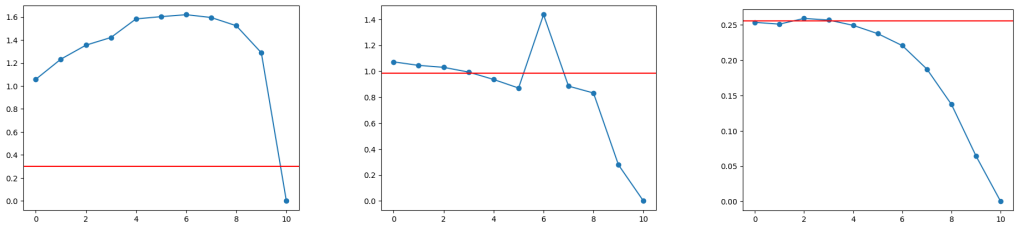

Figure 5.6: DHA-Total persistence in H1 applied on subjects 26, 29, 31.

It is clear that the three outliers once again show distinct trends from other observations. If we choose the PC that can produce the largest total persistence when removing in each hierarchy, then it is probably to choose some extreme values. Such as some extremely large values. For example, subjects 26, 29. We can clearly see that *subject* 26 has a lot of extreme large values, while *subject* 29 suddenly has an abnormally large value when $x = 6$ (that is, when 7 PCs have been removed), making the trend of the curve is cut off directly from the middle, forming a mountain-shaped image. In addition, subjects 26, 29 also had a sudden drop. Unlike the six pictures in the previous normal class, when subject 26 and subject 29 began to show a downward trend, their decline speed was very rapid, instead of having a smooth and smooth downward trend like the previous six figures. And *subject* 31 belongs to the special one among these three outliers. In fact, according to the results of this section and the previous sections of this chapter. *Subject* 31 may be similar in structure to the observation of the normal class. The reason why *subject* 31 is an outlier is probably because the scale of values is not at the same level as other observations. The curve starts from 0.25 and then went down.

5.6 Results of combined DHA based on persistence entropy

In the last section, we study the topological changes of the data set directly through persistence entropy, and we also indirectly use all the previously mentioned methods as auxiliary elements.

For this data set, we investigate two problems:

1. At what dimension to obtain the largest possible persistence entropy, and which principal components support these dimensions.

2. In what dimension, we can ensure that many unnecessary dimensions (that is, principal components) are removed, and at the same time, we can ensure that the persistence entropy of the removed persistence diagram has no

significant change compared with the original one. And determine which principal components are retained.

In order to solve these two problems, we used two analysis methods, DHA-Maximum persistence entropy and DHA-Approximate original entropy, respectively. At the same time, we also use the trajectory obtained in the previous two sections to construct auxiliary curves for us. For more details and specific ways of how works this *combined DHA* analysis method, please consult in the section 4.5.

As the previous section, we exhibit three results of the normal class, there are subjects $30, 32, 33$ and they are in H_1 because the reason explained above.

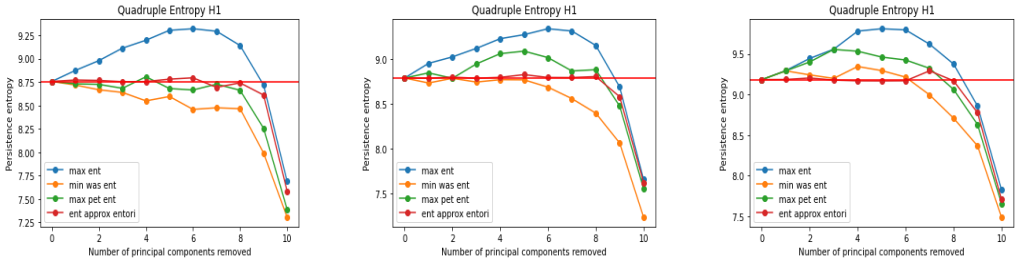

Figure 5.7: Combined DHA in H1 applied on subjects 30, 32, 33.

These figures are called *Quadruple entropy plot*. Because all those curves represent the variation of persistence entropies and *Quadruple* means they were calculated through four different trajectories.

Unlike the figures in the previous two sections of this chapter, the graphics in this section with $x = j$ means j PCs have been removed. So when $x = 0$ means in that time the persistence entropy belongs to the original diagram. And when $x = 10$ means the post-removal only has two PCs remained. Of course, as we said several times before, their is no need to show the result when $x = 11$ because H_1 could not birth with just one dimension left.

Obviously, we can see that these three subjects have almost the same behaviors in the orange curves generated by the trajectory of DHA-Wasserstein distance. These curves start from the original entropy, and then basically keep monotonically decreasing. The trend of orange curves also make sense. Because we can see in the sections 4.2 and 5.4. If we remove PCs according to the trajectory order in DHA-Wasserstein distance, we find that the generated post-removal diagrams will be farther and farther away from the original diagram, and the increase in this distance is monotonous before reaching the peak. On the other hand, we can also see that the blue curves of these three pictures are the curves generated when PCs are removed according to the trajectory order in DHA-Maximum entropy. They also showed a high degree of similarity. They all start from the original entropy,

and then because the maximum value is selected every time, it increases all the way up to the peak, and then declines smoothly. And the other three curves are all within the range delineated by the blue curve, and none of the curves can pass through the blue curve and go beyond this range.

Because the green curve is the curve generated when PCs are removed according to the trajectory order in DHA-Total persistence. The similarity between itself and DHA-persistence entropy is relatively high. So we will not explain it here in particular. And look directly at the red curve, which is the curve generated when PCs are removed according to the trajectory order in DHA-Approximate original entropy. We can see that this curve closely fits the red horizontal line representing original entropy in most hierarchies, until $x = 7$ or $x = 8$, there is a drastic change and it breaks away from the red horizontal line.

According to our previous definition of topological latent dimension (definition 3.25). We can consider an interval of tolerance that does not include $x = 10$ (because when $x = 10$ always present a rapid fall), and then find that the value that best matches the definition of latent topological dimension will be around $dim = 4$ ($x = 8$, we removed 8 PCs from all 12 PCs).

Finally we also observe that by the time we get to the last few PCs, all four curves have basically converged and coincided with the same curve.

And then we observe the results of the three outliers in the quadruple entropy plot.

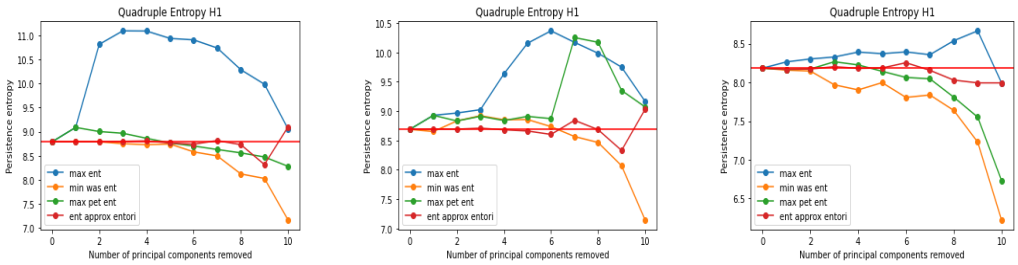

Figure 5.8: Combined DHA in H1 applied on subjects 26, 29, 31.

First we observe the blue curves, the blue curve of *subject* 26 reaches the peak when $x = 2$, that is to say, after we only remove two PCs in the order of the trajectory of DHA-Maximum entropy, we immediately reached the possible maximum value of persistence entropy. For *subject* 29, the change of its blue curve is too drastic. When $x = 3$, the blue curve suddenly rises sharply, that is to say, there is a very large topological difference between the PCs selected to be removed after this hierarchy and the two PCs removed when $x = 1$ and $x = 2$. *Subject* 31 is because its blue curve is not significantly higher than the other three curves, and

it has just reached the peak until $x = 9$.

Another curve we need to look at is the green one. We noticed that the green curve of *subject* 26 does not maintain the same upward and downward trend as the blue curve as the three green curves in the normal class above. Instead, it fits the red curve of original entropy very well and decreases monotonically after $x = 2$. The green curve for *subject* 29 is even more unusual. When thinking about $x = 6$, the green curve suddenly skyrocketed, and then showed an inexplicable trend. The green curve of *subject* 31 is the same as the three examples in the normal class.

The orange curves for all three subjects are in line with our expectations and show no abnormalities. It is possible that the order given by the trajectory generated by the DHA-Wasserstein distance can always provide us with a form that allows the persistence entropy to decline smoothly.

Finally we look at the red curve. We noticed that the red curves of *subject* 26 and 29 both went up abnormally when $x = 9$. And when $x = 10$, its value coincides with the blue curve. Although the last four curves converged and overlapped together in the three pictures in the previous normal class. But the way the red curve coincides with the blue curve is still very strange. However, *subject* 31 does not develop downward along with the orange and green curves, but coincides almost horizontally with the blue curve and $x = 10$.

Similarly, we find that for *subject* $26, 29, 31$. These four curves did not converge to one curve at the end, but diverged completely and irregularly.

According to the results we obtained, subject 26 has the topological latent dimension 8 (with H_1); subjects 27 also has 8, and subjects 31 has 7.

Another notable result occurs in H_0. We give three examples here. We exhibit subjects $25, 29, 32$ to show that this abnormality has nothing to do with whether it is an outlier.

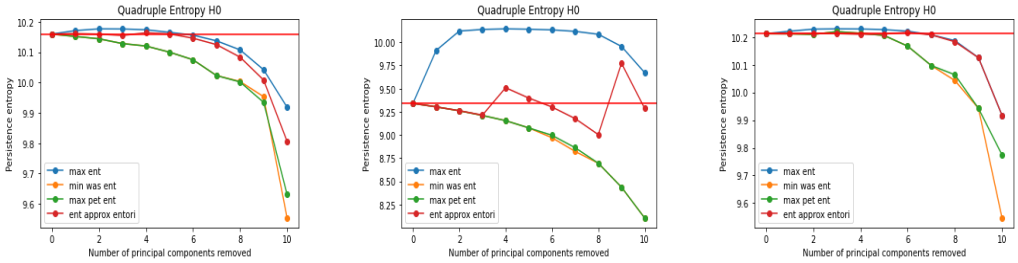

Figure 5.9: Combined DHA in H0 applied on subjects 25, 29, 32.

We found that in H_0, the green curve shows a high correlation with the orange curve. This may be due to the fact that for the persistence diagrams of H_0, in each hierarchy, the PC that has the least impact on the overall is also able to retain the

maximum total persistence each time. In other words, in the persistence diagram of H_0, the Wasserstein distance between the total persistence and post-removal diagram and the original diagram is highly correlated.

And we can also find that the topological latent dimensions (with H_0) for subjects $25, 32$ are clear, they are 6 and 5 (because of $x = 6$ and $x = 7$ respectively) and for subject 29 we cannot decide because the sudden changes in $x = 4$, $x = 8$ and $x = 9$ of the red curve.

Chapter 6

Conclusion

The core purpose of this work is based on the validity of the Manifold Hypothesis, since we attempt to find a latent dimension of a given data set. The latent dimension can be different depending on the criteria on which it is based. Thus we defined a topological latent dimension (Definition 3.25) in Chapter 3 and then we implemented a method to determine it. To accomplish our purpose, we designed some key analytical methods for the first time, such as Directed Hierarchy Analysis (DHA), in Chapter 4.

Based on the PCA method, we concentrated the original data information into a relatively smaller number of principal components than in the original dataset. Then we used persistent homology and persistence entropy from algebraic topology, and some other auxiliary methods (such as Wasserstein distance); we combined them with the DHA analysis method, and then conducted research. In addition to finding a topological latent dimension through this method, we also found which PCs are topological latent variables through the trajectory (Definition 4.2) generated by the DHA method.

In Chapter 5, we first performed PCA processing on the data set (5.1) and obtained 12 PCs. Then we analyzed the 11 study subjects using the combined DHA method. We successfully found outliers through DHA-maximum entropy —they are subjects 26, 29, and 31. By focusing on H_1, we successfully found that the topological latent dimension of most observations is dim $= 4$. We also found a strong correlation between total persistence and Wasserstein distance with H_0.

Our method also has disadvantages. The information contained in persistent entropy is implicit, and if there is no analysis method based on the amount of topological information (for example, the topological classifiers used in [3]), the topological latent dimension we have found is difficult to be used for other purposes. Therefore, this work provides a theoretical basis for an analysis method based on the amount of topological information that may appear in the future.

Bibliography

[1] Edelsbrunner, Letscher, and Zomorodian, *Topological persistence and simplification*, Discrete & Computational Geometry **28** (2002), 511–533.

[2] Herbert Edelsbrunner, *Persistent homology: theory and practice*, (2013).

[3] Aina Ferrà, Gloria Cecchini, Fritz-Pere Nobbe Fisas, Carles Casacuberta, and Ignasi Cos, *A topological classifier to characterize brain states: When shape matters more than variance*, arXiv preprint arXiv:2303.04231 (2023).

[4] Werner H Greub, *Linear Algebra*, vol. 23, Springer Science & Business Media, 2012.

[5] Allen Hatcher, *Algebraic Topology*, 2005.

[6] Matteo Rucco, Filippo Castiglione, Emanuela Merelli, and Marco Pettini, *Characterisation of the idiotypic immune network through persistent entropy*, Proceedings of ECCS 2014: European Conference on Complex Systems, Springer, 2016, pp. 117–128.

[7] Claude E Shannon, *A mathematical theory of communication*, The Bell System Technical Journal **27** (1948), no. 3, 379–423.

[8] Afra Zomorodian and Gunnar Carlsson, *Computing persistent homology*, Proceedings of the twentieth annual symposium on computational geometry, 2004, pp. 347–356.

Annex 1: Results for all subjects

In the first annex, we exhibit all the results we have obtained using the *combined DHA* in H_1, H_0, DHA-Total persistence in H_1 and DHA-Wasserstein distance in H_1.

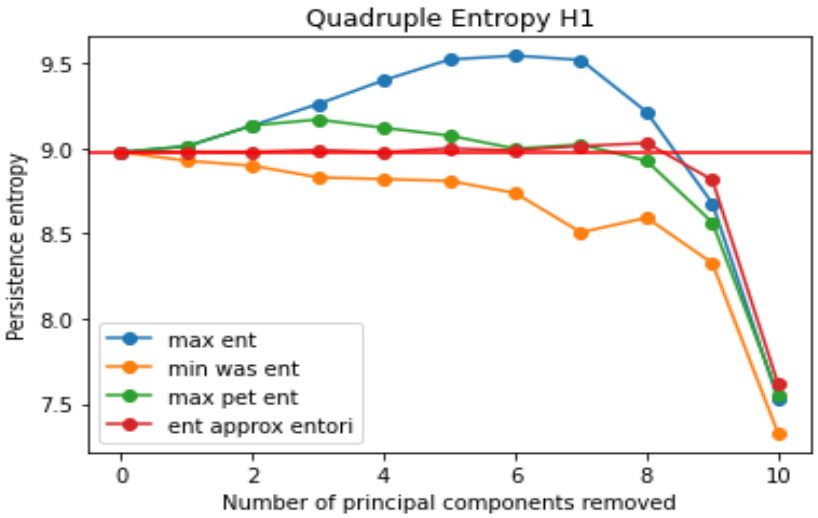

Figure 1: Combined DHA in H1 applied on subject 25.

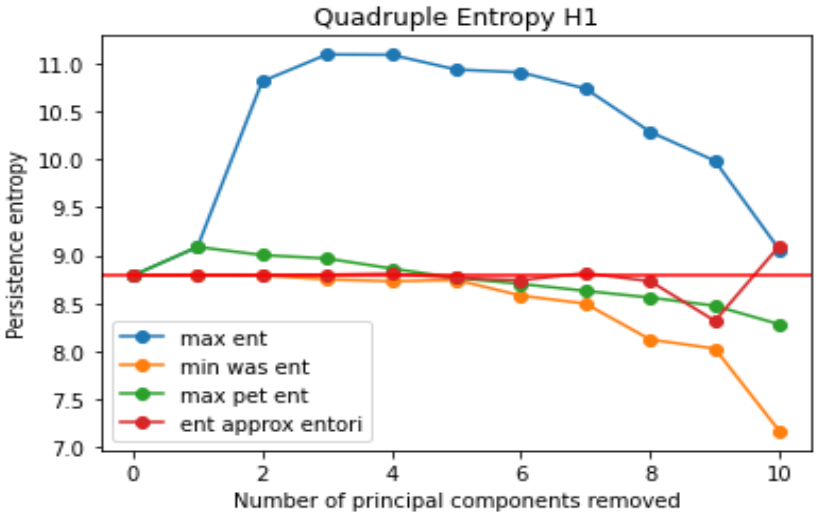

Figure 2: Combined DHA in H1 applied on subject 26.

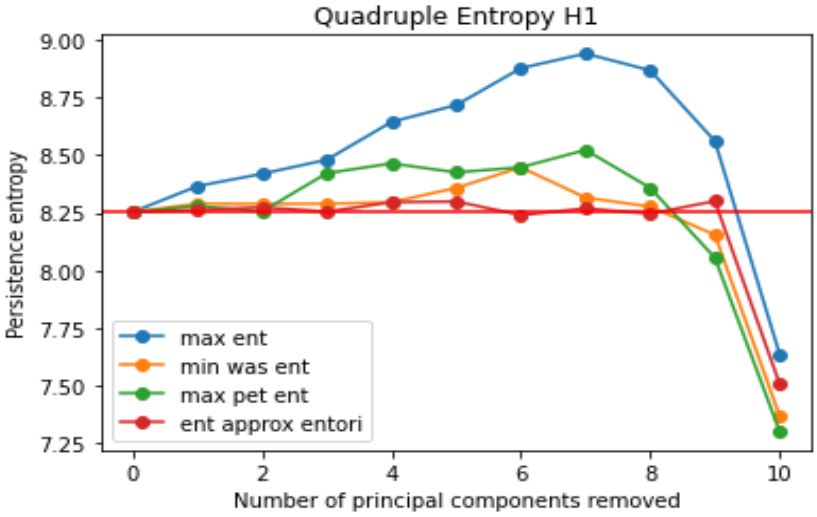

Figure 3: Combined DHA in H1 applied on subject 27.

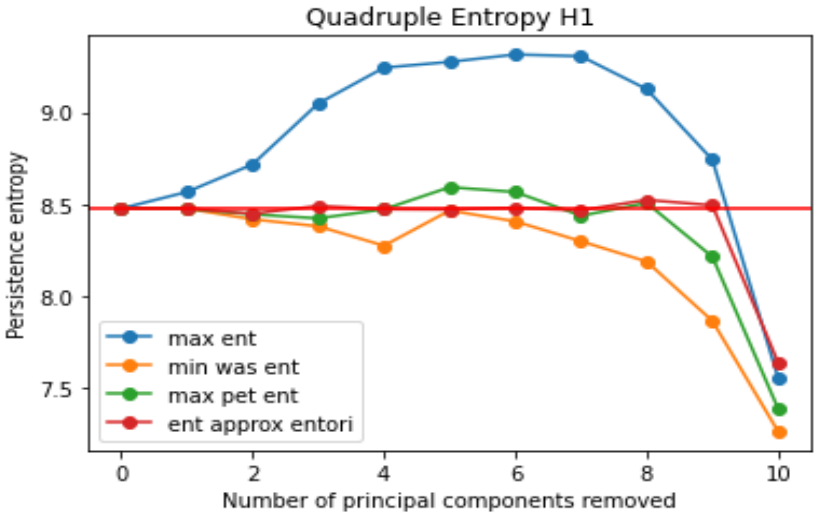

Figure 4: Combined DHA in H1 applied on subject 28.

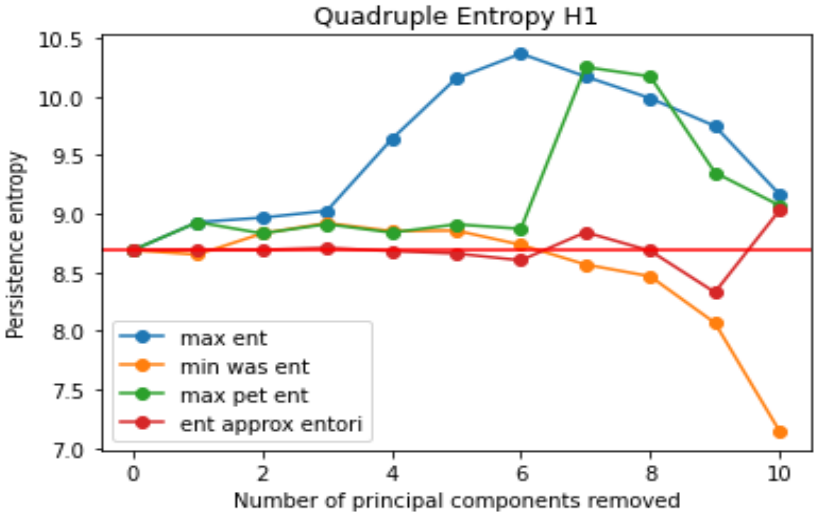

Figure 5: Combined DHA in H1 applied on subject 29.

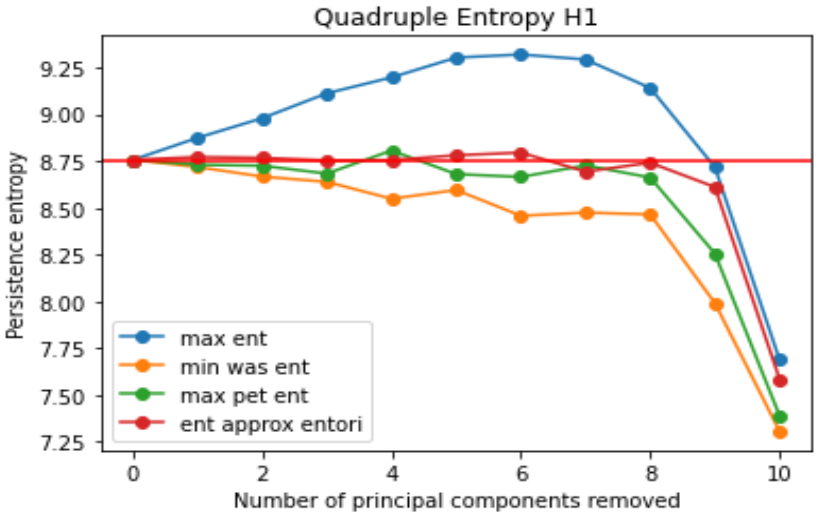

Figure 6: Combined DHA in H1 applied on subject 30.

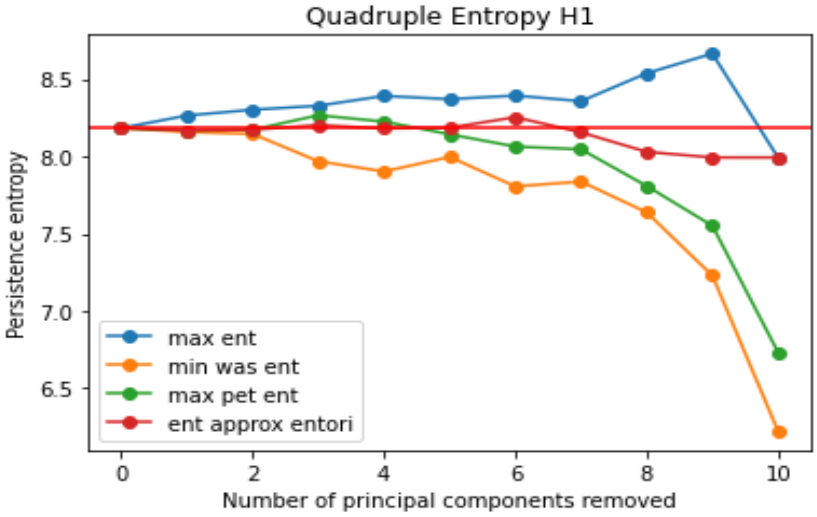

Figure 7: Combined DHA in H1 applied on subject 31.

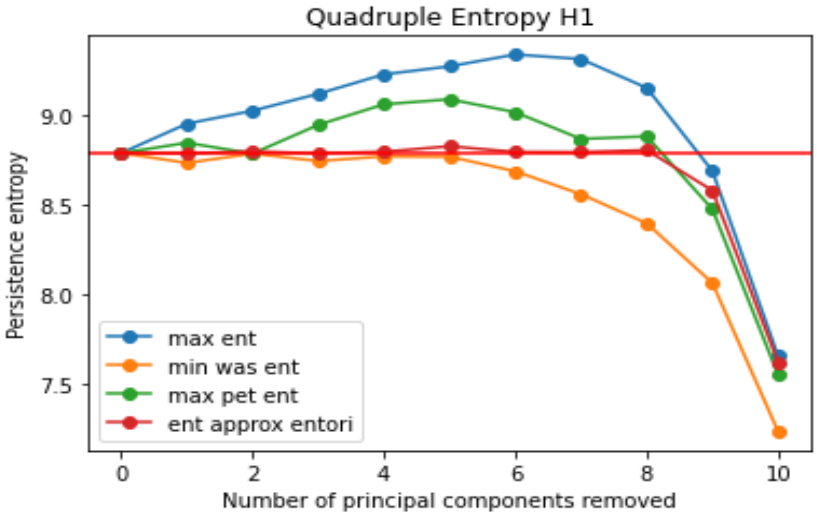

Figure 8: Combined DHA in H1 applied on subject 32.

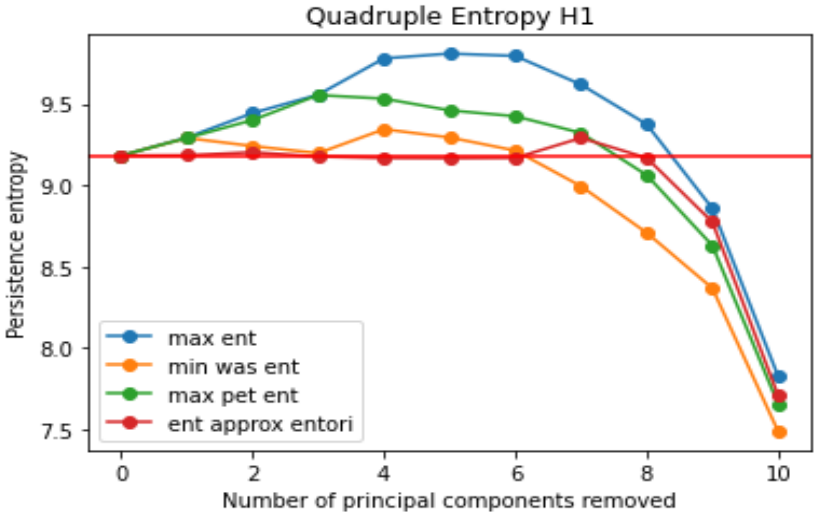

Figure 9: Combined DHA in H1 applied on subject 33.

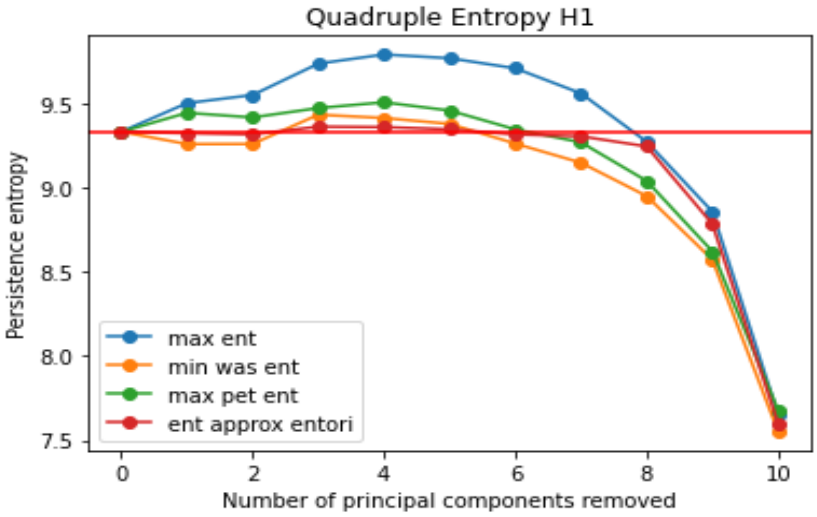

Figure 10: Combined DHA in H1 applied on subject 34.

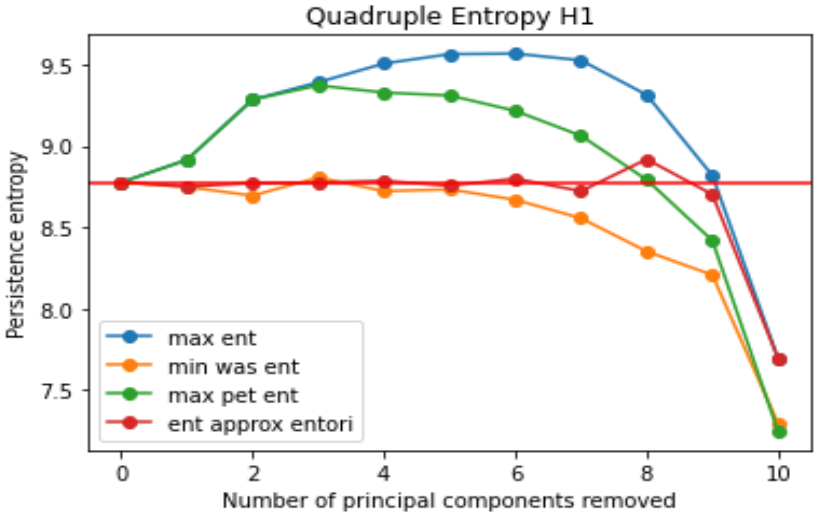

Figure 11: Combined DHA in H1 applied on subject 35.

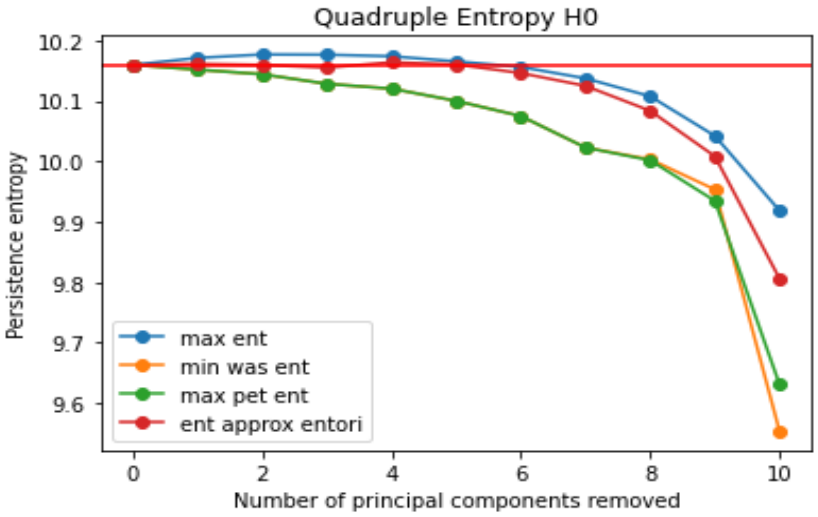

Figure 12: Combined DHA in H0 applied on subject 25.

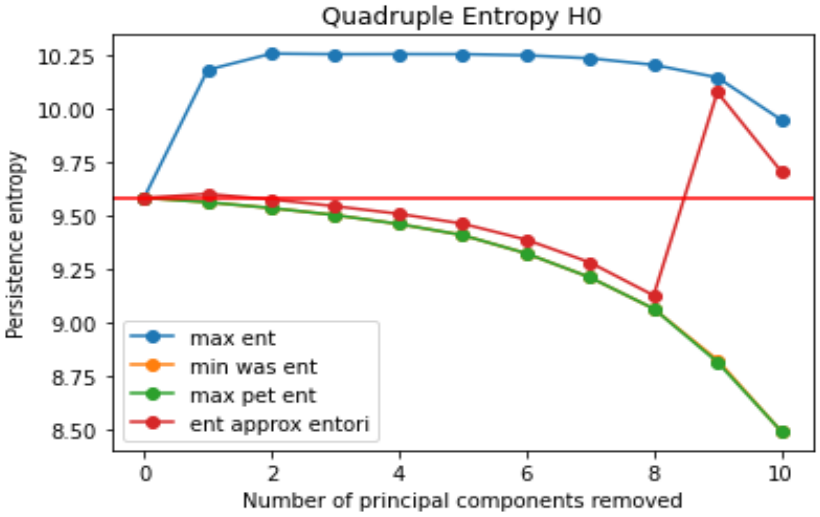

Figure 13: Combined DHA in H0 applied on subject 26.

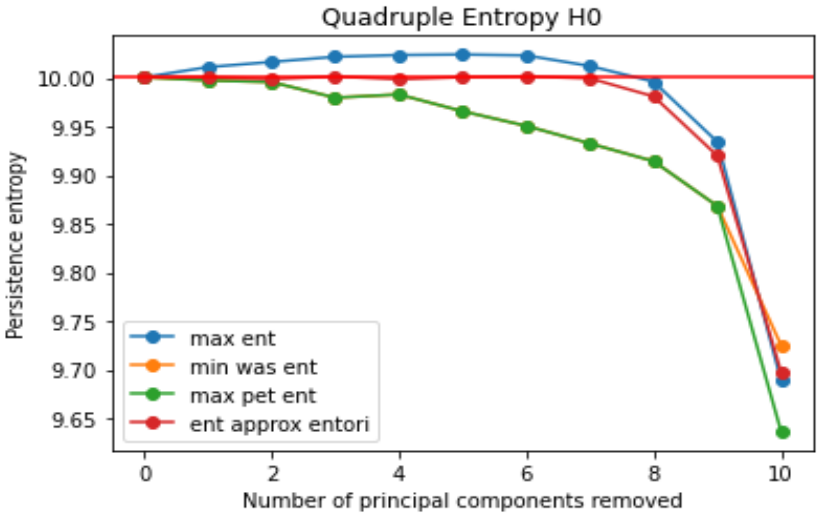

Figure 14: Combined DHA in H0 applied on subject 27.

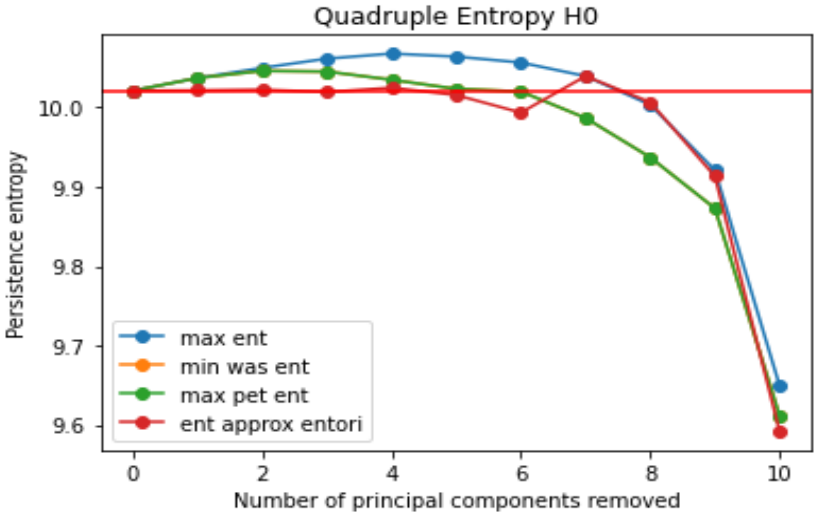

Figure 15: Combined DHA in H0 applied on subject 28.

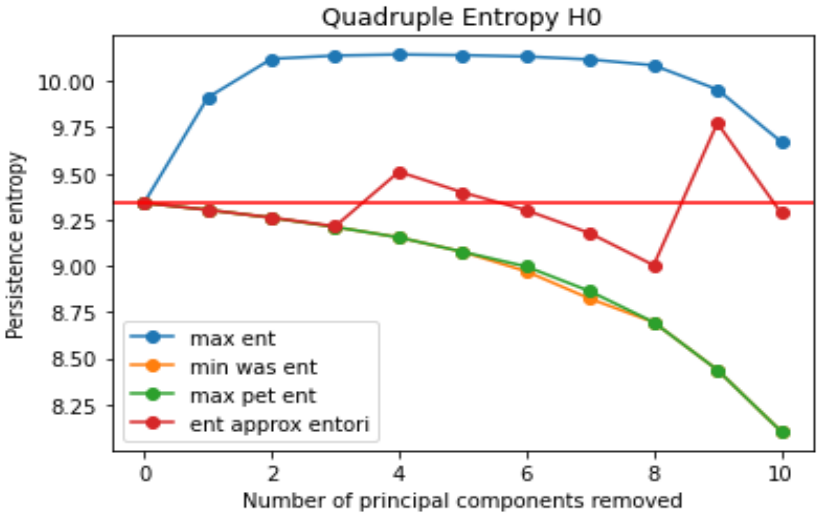

Figure 16: Combined DHA in H0 applied on subject 29.

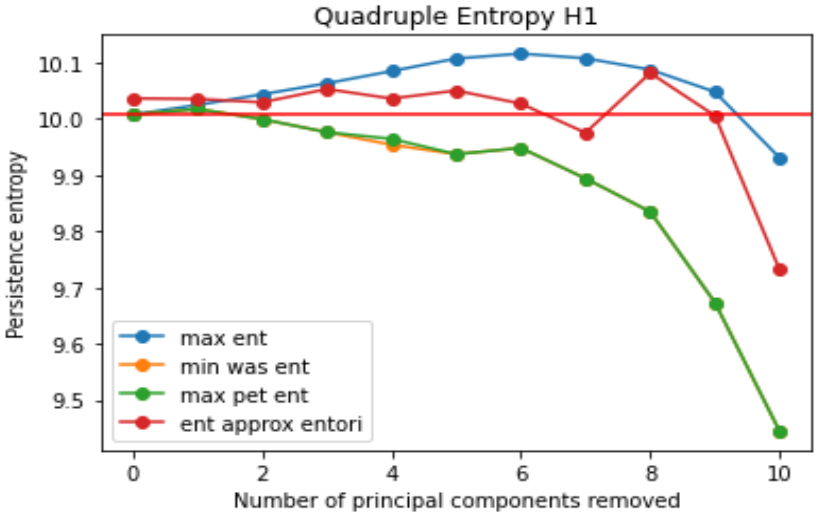

Figure 17: Combined DHA in H0 applied on subject 30.

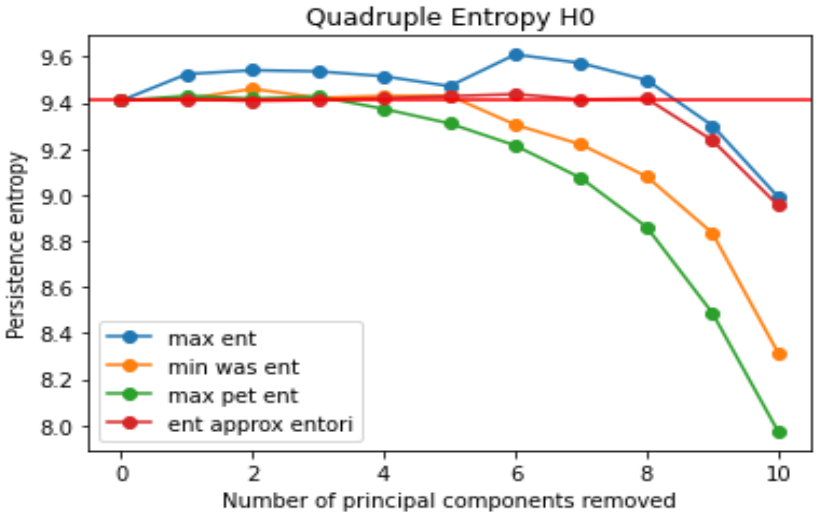

Figure 18: Combined DHA in H0 applied on subject 31.

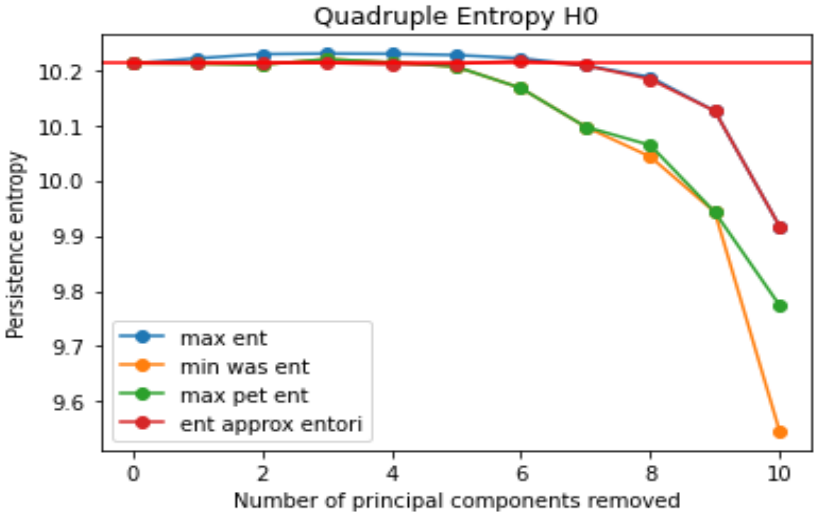

Figure 19: Combined DHA in H0 applied on subject 32.

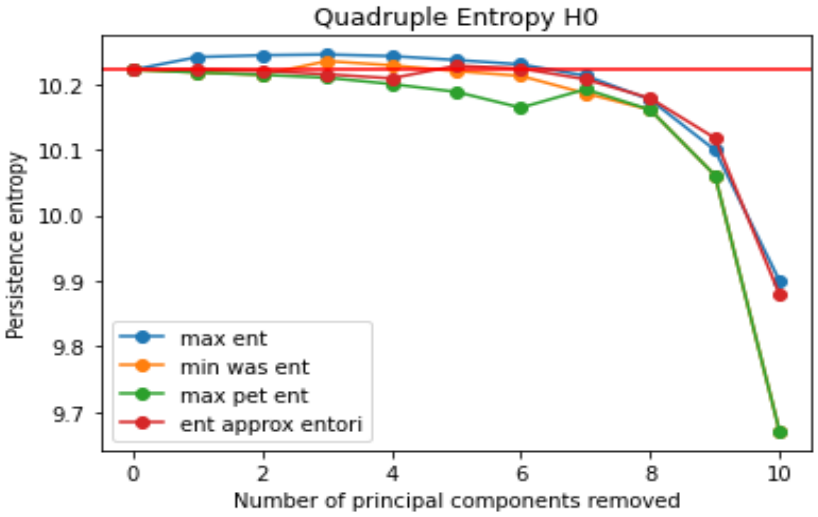

Figure 20: Combined DHA in H0 applied on subject 33.

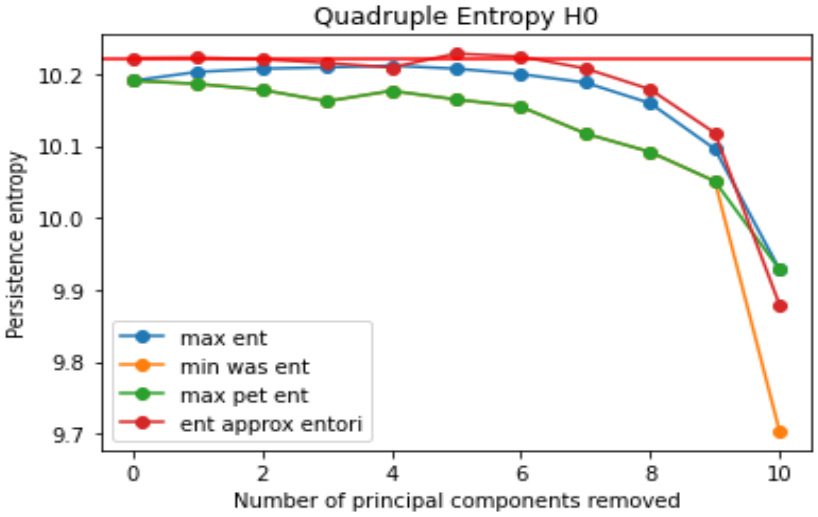

Figure 21: Combined DHA in H0 applied on subject 34.

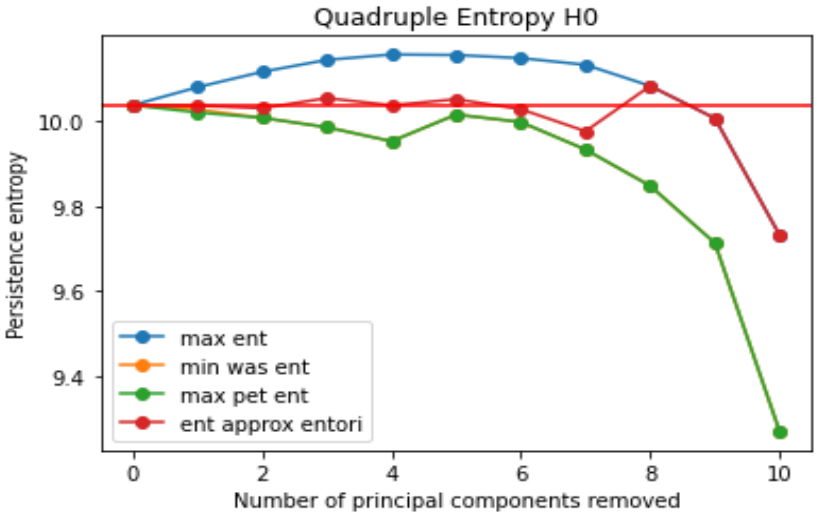

Figure 22: Combined DHA in H0 applied on subject 35.

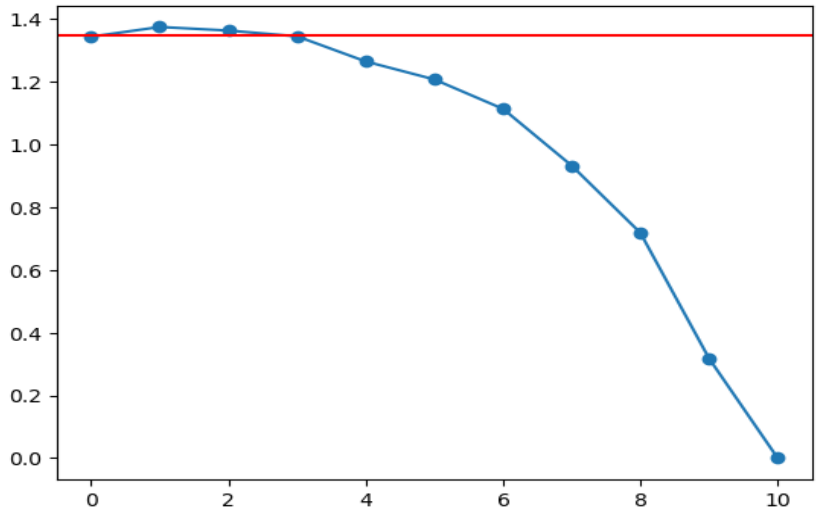

Figure 23: DHA-Total persistence in H1 applied on subject 25.

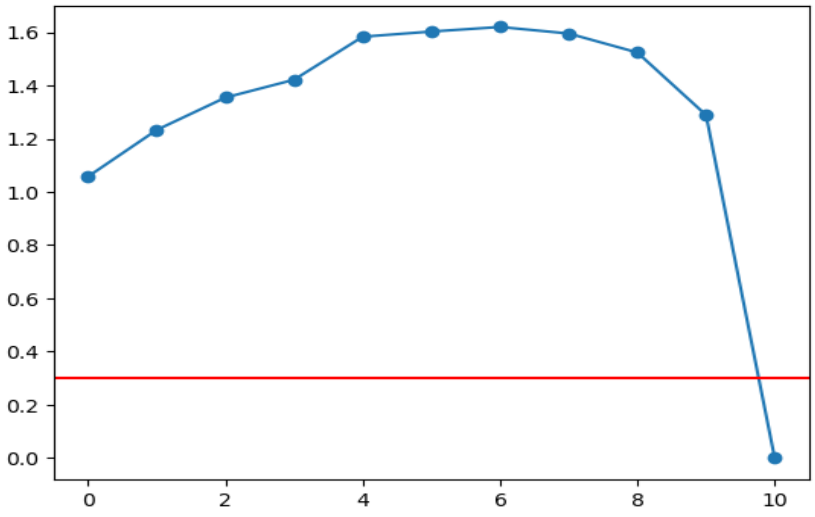

Figure 24: DHA-Total persistence in H1 applied on subject 26.

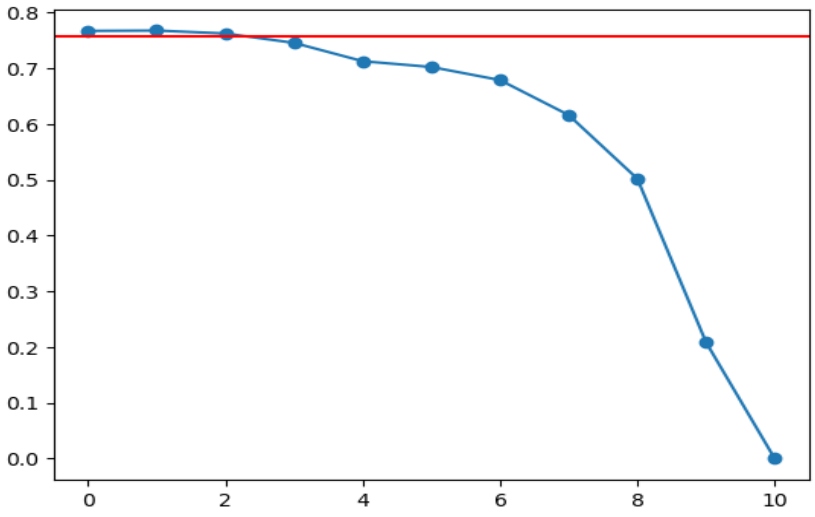

Figure 25: DHA-Total persistence in H1 applied on subject 27.

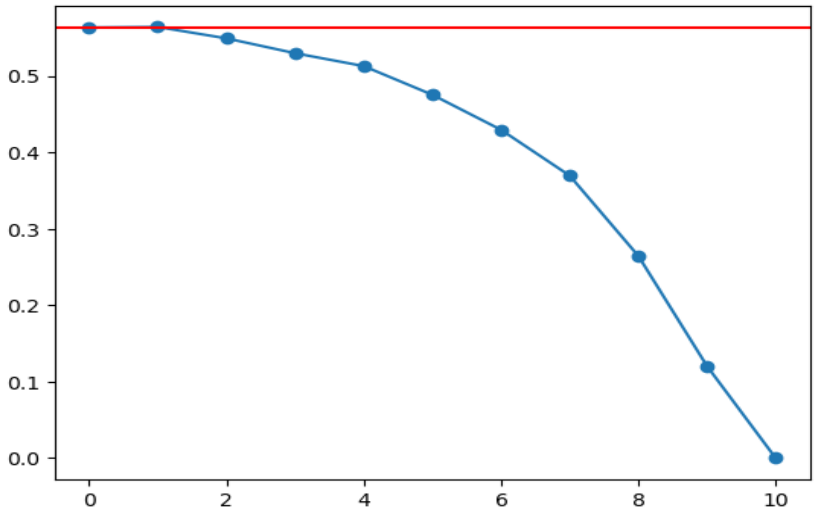

Figure 26: DHA-Total persistence in H1 applied on subject 28.

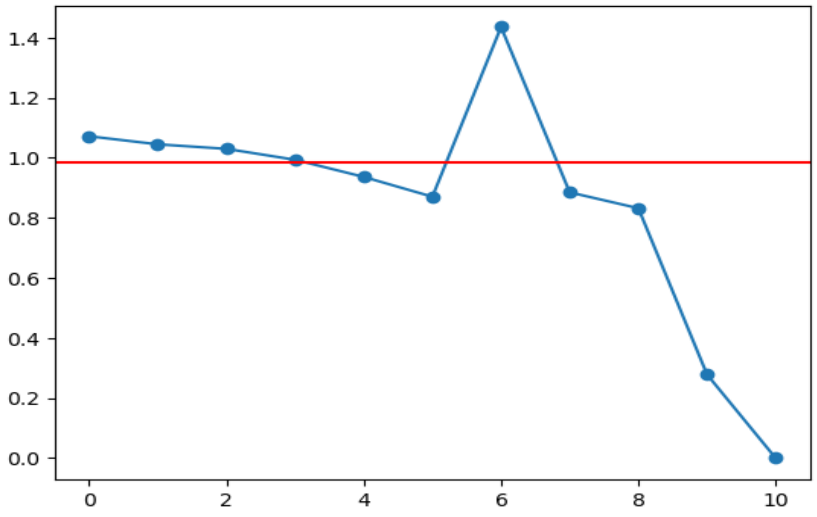

Figure 27: DHA-Total persistence in H1 applied on subject 29.

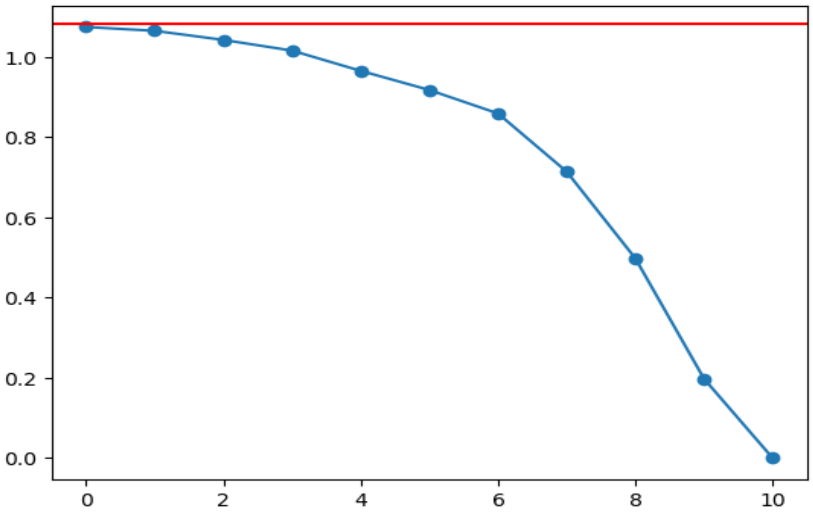

Figure 28: DHA-Total persistence in H1 applied on subject 30.

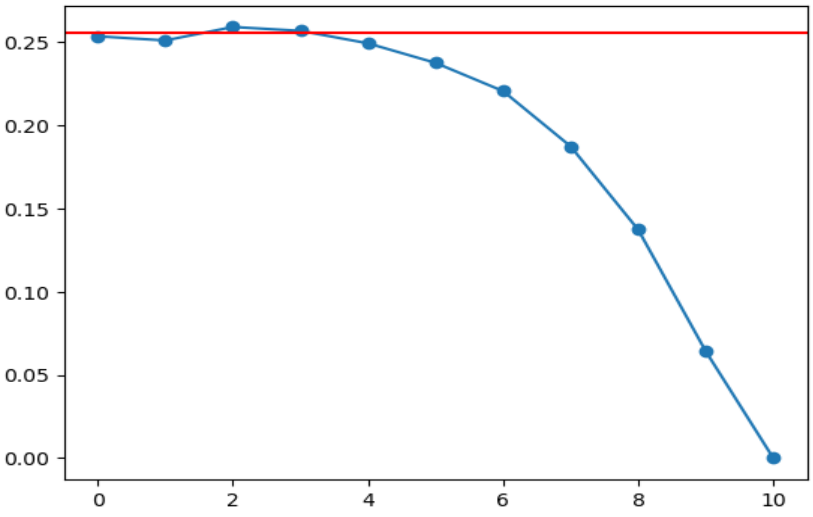

Figure 29: DHA-Total persistence in H1 applied on subject 31.

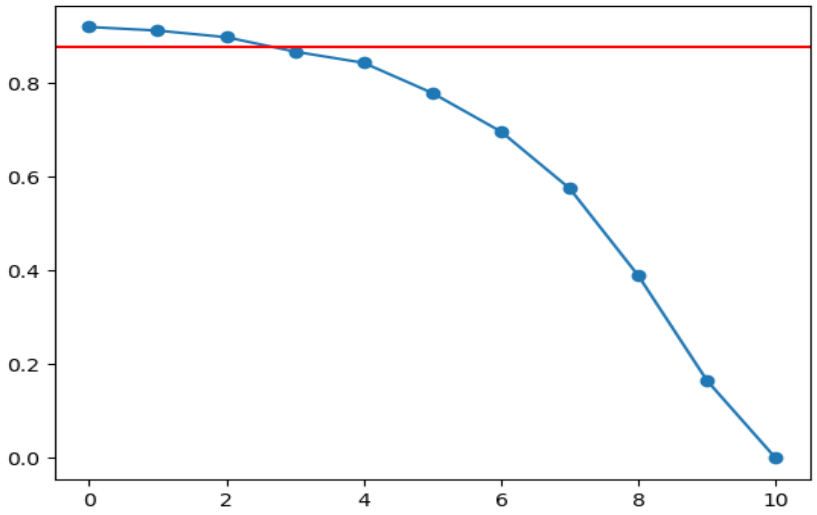

Figure 30: DHA-Total persistence in H1 applied on subject 32.

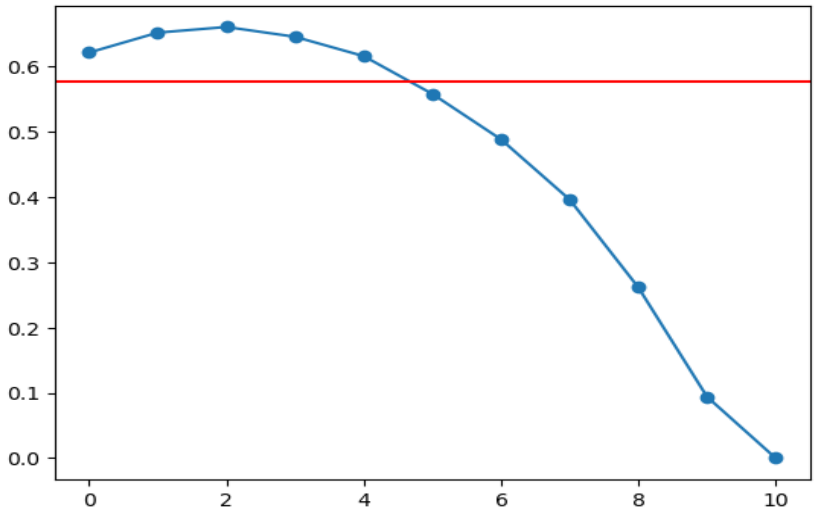

Figure 31: DHA-Total persistence in H1 applied on subject 33.

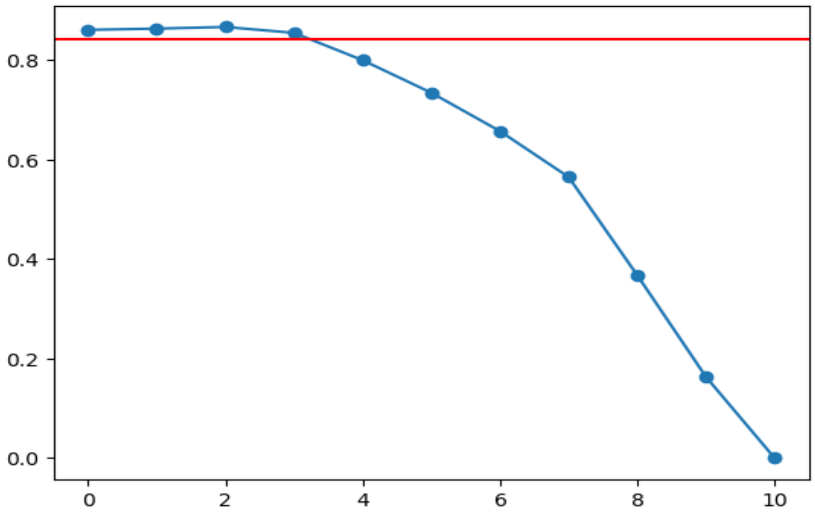

Figure 32: DHA-Total persistence in H1 applied on subject 34.

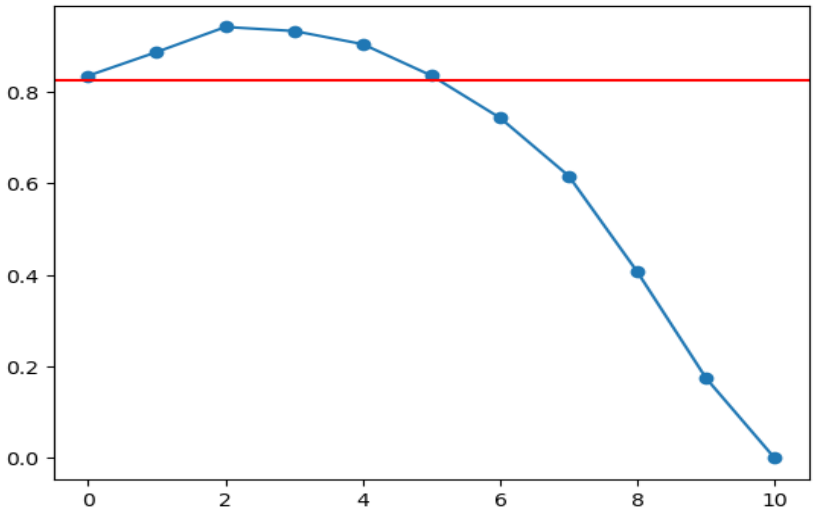

Figure 33: DHA-Total persistence in H1 applied on subject 35.

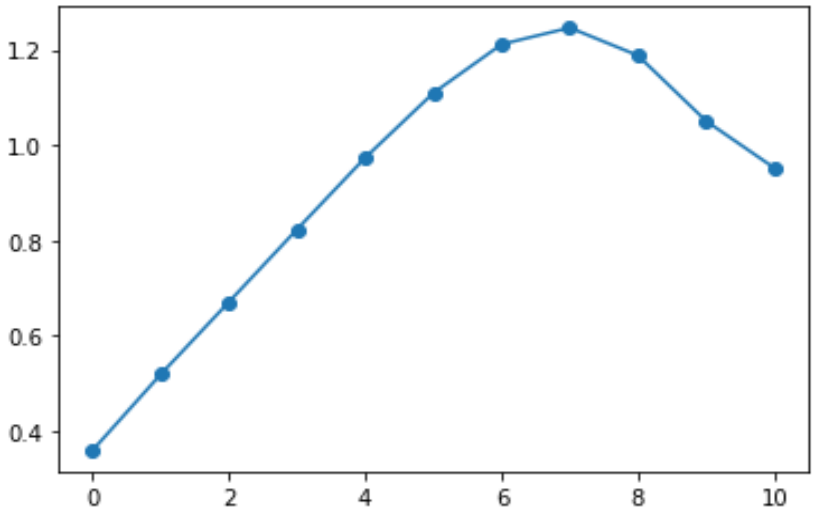

Figure 34: DHA-Wasserstein distance in H1 applied on subject 25.

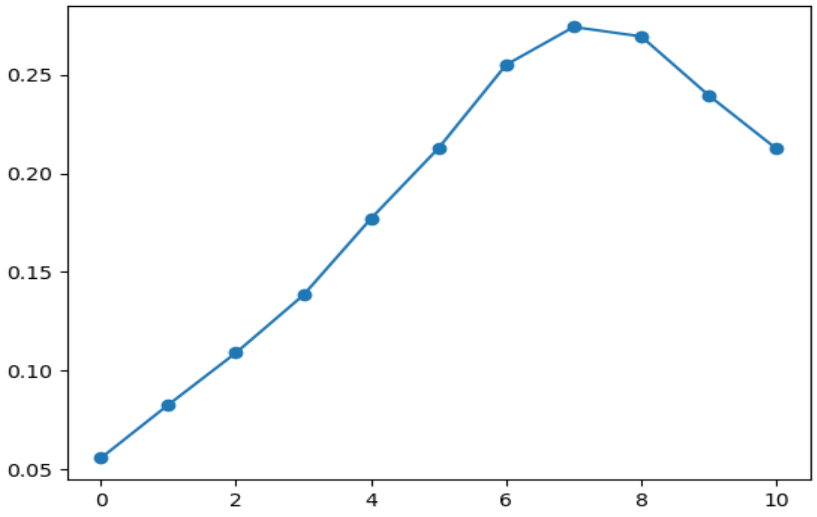

Figure 35: DHA-Wasserstein distance in H1 applied on subject 26.

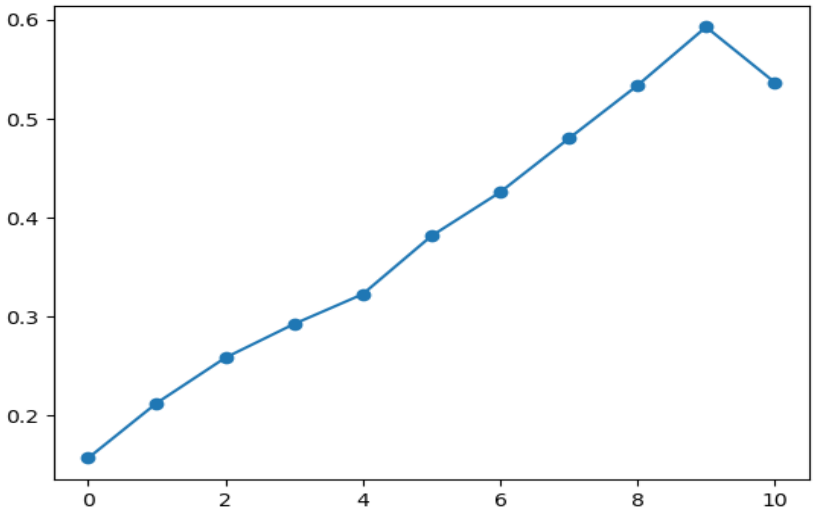

Figure 36: DHA-Wasserstein distance in H1 applied on subject 27.

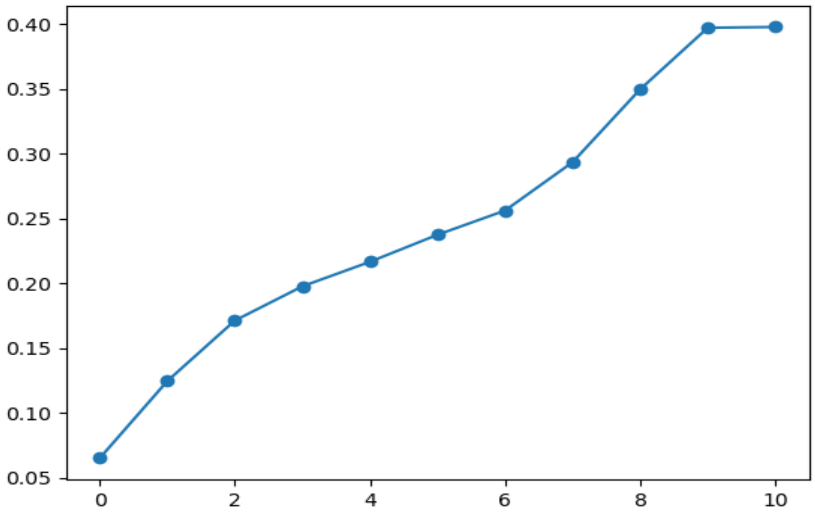

Figure 37: DHA-Wasserstein distance in H1 applied on subject 28.

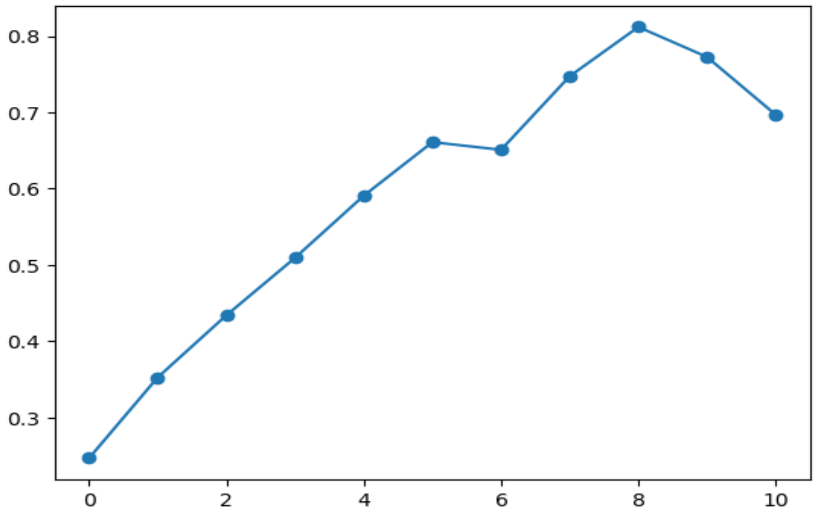

Figure 38: DHA-Wasserstein distance in H1 applied on subject 29.

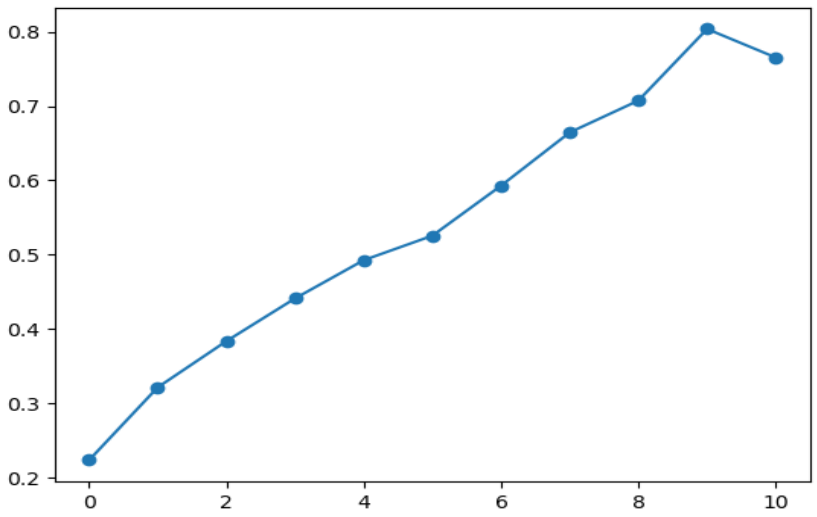

Figure 39: DHA-Wasserstein distance in H1 applied on subject 30.

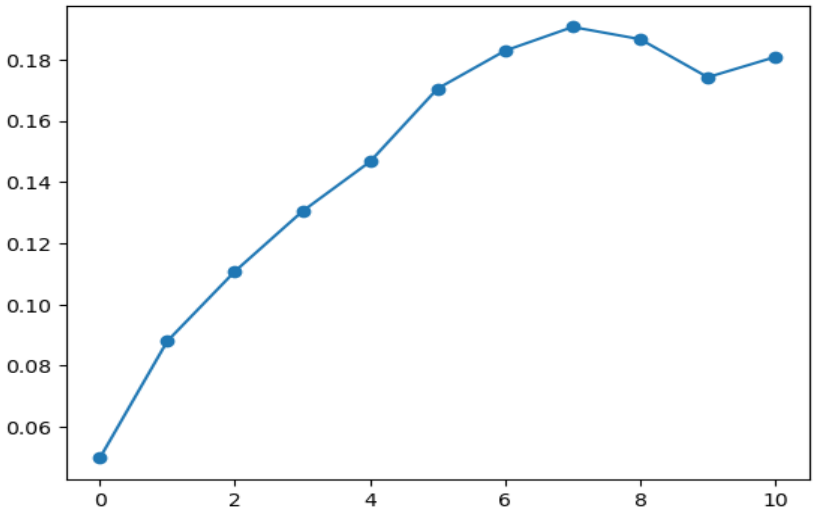

Figure 40: DHA-Wasserstein distance in H1 applied on subject 31.

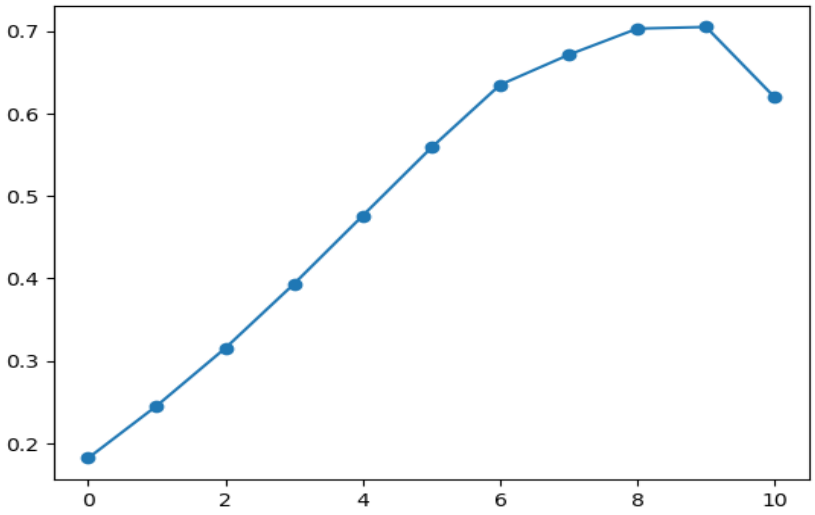

Figure 41: DHA-Wasserstein distance in H1 applied on subject 32.

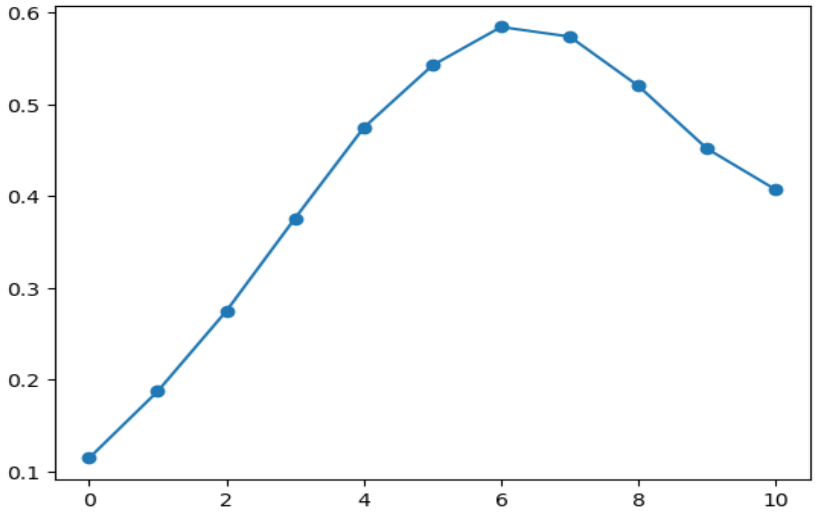

Figure 42: DHA-Wasserstein distance in H1 applied on subject 33.

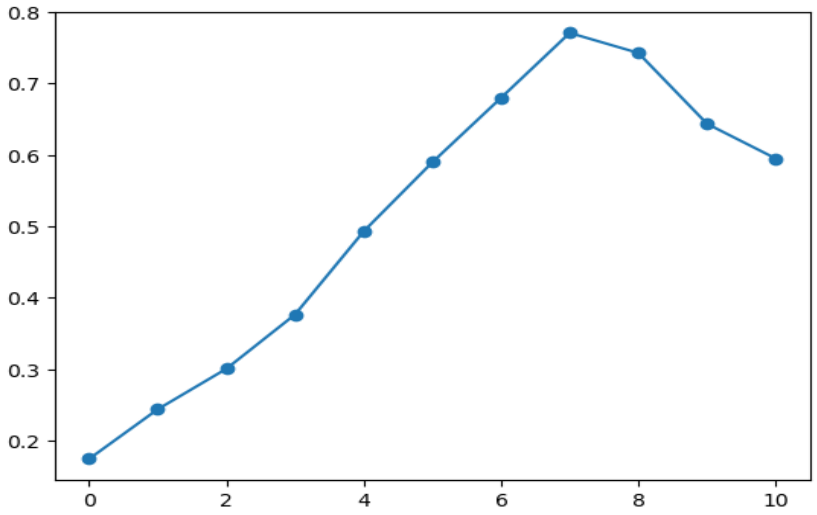

Figure 43: DHA-Wasserstein distance in H1 applied on subject 34.

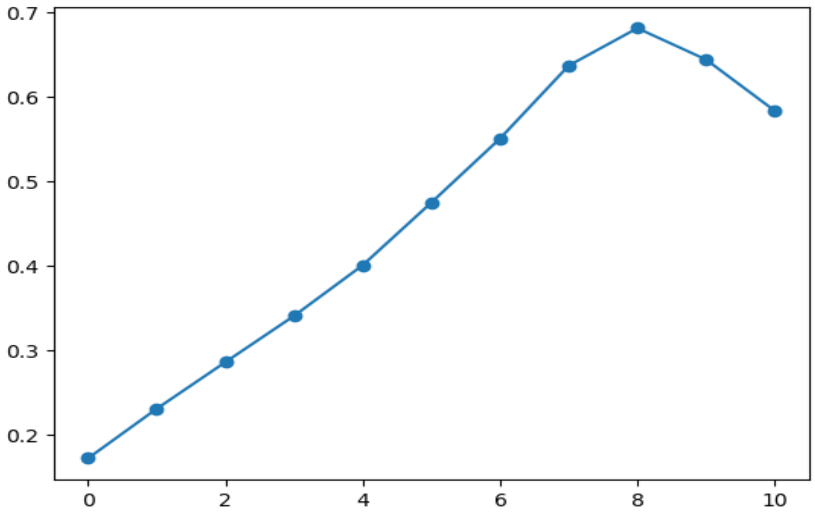

Figure 44: DHA-Wasserstein distance in H1 applied on subject 35.

Annex 2: Code used for this work

We also put the code we have used to accomplish the algorithm defined by us. Here we just show the core code what is the function for DHA (Also called AJD in spanish). There will be some comments in chinese or spanish to make me not forget.

```
"""

Desde aqui, es  Análisis de jerarquía dirigido
从此处开始，下面的就是被我称为针对性层次分析(AJD)的方法

"""

#distance_dir = np.zeros([12,2],float)

def ajdcui(data,w,c,diagori,dist):

    z=np.array(range(data.shape[1]))

    x=data.drop(data.columns[w],axis = 1)

    z=np.delete(z,w)

    distance_dir = np.zeros([len(z),2],float)

    if(dist=='ent' and c==0):

        inflist=[]

        for s in range(len(np.isinf(diagori[0][:,1]))):

            if (np.isinf(diagori[0][s,1])==True):

                inflist.append(s)


        for s in range(len(inflist)):

            diagori[0]=np.delete(diagori[0],inflist[s],axis=0)

    if(dist=='pet' and c==0):

        inflist=[]
```

Figure 45: First part of DHA.

```
if(dist=='pet' and c==0):

    inflist=[]

    for s in range(len(np.isinf(diagori[0][:,1]))):

        if (np.isinf(diagori[0][s,1])==True):

            inflist.append(s)

    for s in range(len(inflist)):

        diagori[0]=np.delete(diagori[0],inflist[s],axis=0)


for i in range(len(z)):

    datos_i = x.drop(x.columns[i],axis = 1)
    diagrams_i = ripser(datos_i)['dgms']

    if(dist=='was'):

        if(c==0):

            distance = was(diagrams_i[0], diagori[0])

        elif(c==1):

            distance = was(diagrams_i[1], diagori[1])

    elif(dist=='btn'):

        if(c==0):

            distance = btn(diagrams_i[0], diagori[0])

        elif(c==1):

            distance = btn(diagrams_i[1], diagori[1])
```

Figure 46: Second part of DHA.

```
elif(dist=='pet'):
    if(c==0):
        inflist=[]
        for s in range(len(np.isinf(diagrams_i[0][:,1]))):
            if (np.isinf(diagrams_i[0][s,1])==True):
                inflist.append(s)
        for s in range(len(inflist)):
            diagrams_i[0]=np.delete(diagrams_i[0],inflist[s],axis=0)
        distance = (diagrams_i[0][:,1]-diagrams_i[0][:,0]).sum()
    if(c==1):
        distance = (diagrams_i[1][:,1]-diagrams_i[1][:,0]).sum()
elif(dist=='ent'):
    if(c==0):
        inflist=[]
        for s in range(len(np.isinf(diagrams_i[0][:,1]))):
            if (np.isinf(diagrams_i[0][s,1])==True):
                inflist.append(s)
        for s in range(len(inflist)):
            diagrams_i[0]=np.delete(diagrams_i[0],inflist[s],axis=0)
        per_total = (diagrams_i[0][:,1]-diagrams_i[0][:,0]).sum()
        entropia = 0
        for j in range(diagrams_i[0].shape[0]):
```

Figure 47: Third part of DHA.

```
        for j in range(diagrams_i[0].shape[0]):

            entropia=entropia-((diagrams_i[0][j,1]-diagrams_i[0][j,0])
                               /per_total)*log2((diagrams_i[0][j,1]
                                                 -diagrams_i[0][j,0])
                                                 /per_total)

        distance = entropia

    if(c==1):

        per_total = (diagrams_i[1][:,1]-diagrams_i[1][:,0]).sum()
        entropia = 0
        for j in range(diagrams_i[1].shape[0]):

            entropia=entropia-((diagrams_i[1][j,1]-diagrams_i[1][j,0])
                               /per_total)*log2((diagrams_i[1][j,1]
                                                 -diagrams_i[1][j,0])
                                                 /per_total)

        distance = entropia


elif(dist=='entaproxori'):

    if(c==0):

        inflist=[]

        for s in range(len(np.isinf(diagrams_i[0][:,1]))):

            if (np.isinf(diagrams_i[0][s,1])==True):

                inflist.append(s)

        for s in range(len(inflist)):

            diagrams_i[0]=np.delete(diagrams_i[0],inflist[s],axis=0)

        per_total = (diagrams_i[0][:,1]-diagrams_i[0][:,0]).sum()
        entropia = 0
```

Figure 48: Fourth part of DHA.

```
            for j in range(diagrams_i[0].shape[0]):

                entropia=entropia-((diagrams_i[0][j,1]-diagrams_i[0][j,0])
                                   /per_total)*log2((diagrams_i[0][j,1]
                                                     -diagrams_i[0][j,0])
                                                     /per_total)


            entori=0
            per_ori = (diagori[0][:,1]-diagori[0][:,0]).sum()
            for j in range(diagori[0].shape[0]):

                entori=entori-((diagori[0][j,1]-diagori[0][j,0])
                               /per_ori)*log2((diagori[0][j,1]
                                               -diagori[0][j,0])
                                               /per_ori)


            difference=abs(entropia-entori)

            distance = difference

        if(c==1):

            per_total = (diagrams_i[1][:,1]-diagrams_i[1][:,0]).sum()
            entropia = 0
            for j in range(diagrams_i[1].shape[0]):

                entropia=entropia-((diagrams_i[1][j,1]-diagrams_i[1][j,0])
                                   /per_total)*log2((diagrams_i[1][j,1]
                                                     -diagrams_i[1][j,0])
                                                     /per_total)


            entori=0
            per_ori = (diagori[1][:,1]-diagori[1][:,0]).sum()
            for j in range(diagori[1].shape[0]):

                entori=entori-((diagori[1][j,1]-diagori[1][j,0])
                               /per_ori)*log2((diagori[1][j,1]
                                               -diagori[1][j,0])
                                               /per_ori)


            difference=abs(entropia-entori)

            distance = difference



    colname=int(x.columns.values[i])
    distance_dir[i]=(distance,colname)

if(dist=='btn' or dist=='was'):

    list_distance_dir=distance_dir[:,0].tolist()
    min_dist = list_distance_dir.index(min(distance_dir[:,0]))
    col_min_dist = distance_dir[min_dist,1]

    return [int(col_min_dist),min(distance_dir[:,0])]
```

```
if(dist=='pet'):

    list_distance_dir=distance_dir[:,0].tolist()
    max_dist = list_distance_dir.index(max(distance_dir[:,0]))
    col_max_dist = distance_dir[max_dist,1]

    return [int(col_max_dist),max(distance_dir[:,0])]

if(dist=='ent'):

    list_distance_dir=distance_dir[:,0].tolist()
    max_dist = list_distance_dir.index(max(distance_dir[:,0]))
    col_max_dist = distance_dir[max_dist,1]

    return [int(col_max_dist),max(distance_dir[:,0])]

if(dist=='entaproxori'):

    list_distance_dir=distance_dir[:,0].tolist()
    min_dist = list_distance_dir.index(min(distance_dir[:,0]))
    col_min_dist = distance_dir[min_dist,1]

    return [int(col_min_dist),min(distance_dir[:,0])]
```

Figure 50: Sixth part of DHA.