Análisis topológico de datos: aplicación al reconocimiento de emociones

Guillermo Aguirre Carrazana

Análisis topológico de datos: aplicación al reconocimiento de emociones

Guillermo Aguirre Carrazana

Memoria presentada como parte de los requisitos para la obtención del título de Máster Universitario en Matemática Avanzada por la Universidad de Sevilla.

Tutorizada por

Prof. Rocío González Díaz
Prof. Eduardo Paluzo Hidalgo

Contents

Abstract

Emotion recognition consists of a series of processes to detect human emotions from facial human expressions. Humans interact with each other primarily through speech, but also through body gestures to emphasize a certain part of the conversation and exhibit emotions. The automatic recognition of a person's emotional state has become a very active research field that involves scientists specialized in different areas such as artificial intelligence, computer vision or psychology, among others. Our main objective in this work is to develop a novel approach, based on topological data analysis, for the recognition and classification of emotions that combines features extracted from videos and audios, in their respective spaces of representation, of people expressing emotions.

Resumen

El área de reconocimiento de emociones consiste en una serie de procesos para detectar emociones humanas a partir de expresiones faciales. Los humanos interactuamos entre nosotros utilizando el habla, pero también a través de los gestos corporales para así enfatizar una cierta parte de la conversación y exhibir emociones. El reconocimiento automático del estado emocional de una persona se ha convertido en un campo activo de investigación que involucra científicos especializados en diferentes áreas tales como la inteligencia artificial, la visión por ordenador o la psicología, entre otras. Nuestro principal objetivo en este trabajo es desarrollar un novedoso enfoque, basado en el análisis topológico de datos, para el reconocimiento y clasificación de emociones que combine características extraídas de videos y audios, en sus respectivos espacios de representación, de personas expresando emociones.

1 Introduction

When a person communicates with others, they are constantly sending and receiving nonverbal cues, expressed through body gesture, voice, facial expressions, and physiological changes. If some nonverbal cues coincide with the words that a person says at the moment, they can increase trust, clarity, rapport, and reveal more information than the person's spoken words. If you want to understand people better, it is important to become more sensitive to their body language and nonverbal cues. A particular emotional state produces certain verbal and non-verbal signals, so emotions convey the information regarding personal feelings.

Nowadays, computers are part of our life, but the relationship between a human and a machine is limited. It becomes necessary to know the emotional state of the user to achieve better human-machine cooperation. This way, emotion recognition becomes an important area of research in the fields of computer vision and artificial intelligence due to its important academic potential and social impact application. So far, different approaches have been explored. See, for example [125].

In general, people infer the emotional state of others (such as joy, sadness, or anger) using facial expressions and vocal tones. Assisting in human interaction, as done in the H2020 KRISTINA project[1] (where emotion recognition is applied to help in the interaction between health professionals and migrated patients), allows overcoming linguistic barriers that make communication difficult in healthcare and primary assistance environment. One of the most relevant conclusions that the KRISTINA project reached was that the combination of visual and audio features can develop better predictions than using them separately.

[1] http://kristina-project.eu/en/

In addition to KRISTINA project, other European projects working on the recognition of emotions are H2020 VocEmoApI project[2] and H2020 MixedEmotions project[3].

Works related to emotion recognition have focused on the utilization of various input types such as facial expressions [92, 117, 128], speech [100, 106, 65, 91, 5] and physical signals [57]. Several other emotion classification techniques have been proposed in [131]. Some of them employ prosody contours information of speech to recognize emotions using different classification methods such as, for example, artificial neural networks, the multi-channel hidden Markov model, and the mixture of hidden Markov models. For a further approximation to para-linguistic theory see [108].

A previous work of our group in the field of emotion recognition, [44], developed an approach using only audio signals. Suchwork constitutes the starting point for this master thesis. Specifically, in [44], a model based on topology was developed to obtain a single value for each audio signal. These data were used as input of a support vector machine to classify audio signals into eight different emotions, namely, **neutral, calm, happy, sad, angry, fearful, disgust, and surprised**. The results obtained were close to the existing accuracy for some methods with a greater scope such as [67, 132]. More specifically, speech signals were processed as piece-wise linear functions. Then, a recent tool in the area of Topological Data Analysis (TDA) called persistent entropy, that is the Shannon entropy of persistence barcodes considered as probability distributions [20], was computed from the lower-star filtration obtained from these functions. Such persistent entropy tool summarizes the features that appear in raw signals, as of intensity and intonation. The stability theorem for persistent entropy computed from lower-star filtration [105] obtained from these functions guarantees fair comparison between signals and robustness against noise. Finally, a support vector machine was used to classify emotions via persistent entropy values.

Additionally, it is interesting to analyze how ingredients from TDA as homology, filtration, and persistence that will be explained in the thesis could show up a different vision about data taken from measures of lower dimensions, such as time series. It is not obvious how the data from the time series is transformed into a cloud of higher dimensions. Time-delay embedding proposes to reconstruct the state and dynamics of an unknown dynamical system from measurement or observations of that system taken over time. We refer the reader to the standard text by Kantz and Schreiber [60] for further details. This idea has been employed in a wide variety of contexts including time series modeling [51], closure modeling [96] and applications in pattern

[2] https://cordis.europa.eu/project/rcn/199804_es.html
[3] https://cordis.europa.eu/project/rcn/194226_es.html

recognition, for example in gait recognition [35]. The objective of this method is to use the tools offered by Topological Data Analysis for the sake of decreasing the noise, measure fault, and non-stationary signals' parameters.

In this work, we build a model supported by the TDA theory to extract topological information from image sequences obtained from videos of people expressing emotions. The idea is to stack images and compute a 3D simplicial complex. We extract eight different features from a video file considering, respectively, the distance to eight fixed planes (2 horizontal, 2 vertical, 2 oblique and 2 depth planes) in order to completely capture the movement in emotion sequence. For each plane π the persistence diagram is calculated as it has been previously done in [68, 69] to gait classification. What is new in this thesis is that we obtain the persistence entropy associated. Putting together all this information, we built a vector associated with each emotional video sequence. This methodology is evaluated using the Video-Dataset from the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [74].

We also propose a method alternative to [44] for representing audio signals obtained from videos of people expressing emotions, by adopting techniques from nonlinear time series analysis [60]. The idea is to extract features from time series and use them as input for a classifier. A novelty of our method is the use of Takens' delay embedding to obtain topological information from the audio signals. As we will see in this thesis, the system requires very little parameter tuning and can be trained with small amounts of data. This methodology is evaluated using the Audio-Dataset extracted from RAVDESS.

Finally, we develop a methodology to combine both topological data computed for the two respective channels (images and audios) obtained from the videos. We then will obtain a topological signature to train a machine learning procedure to classify emotions and will demonstrate that our results outperform state-of-the-art methods.

The thesis is structured in the introduction and six more chapters. Basic emotion theory is introduced in Chapter 2. In Chapter 3, the principal concepts from computational topology, topological data analysis of time series, and machine learning knowledge required for the model are explained. In Chapter 4, the methodology followed on experiments is developed. Results obtained from different training approaches are shown in Chapter 5 together with comparisons with state-of-the-art methods. The chapter 6 focuses on explaining the use of different programming languages and the different libraries applied, as well as discussing the implementation of the completed method. Finally, Section 7 provides conclusions and future work ideas.

2 Biometrics for emotion recognition

Biometrics are biological measurements or physical characteristics used to generate mathematical models of physical (e.g. hand geometry, iris, fingerprints, gait, and so on) and behavioral (e.g. signal, patterns) features to recognize patterns for identification. The biometric details that are different for each person are used as distinct biometric data.

A person's emotional state can influence concentration, task solving, and decision-making skills. Emotional communication occurs through non-verbal channels, like body movements, the tone and pitch of the voice, gestures displayed through body language, and physical distance between the peoples speaking. Expressing and recognizing emotions help us understand the intentions of others, build better relationships, avoid or resolve conflicts better, and move past difficult mental and physical states more easily.

Facial expressions, eye contact, hand gestures, and other cues can transmit more information than just words or language. It can emphasize a message, distract and redirect your attention to highlight details, and influence the course of your actions. Among the non-verbal components that have emotional meaning, facial expressions are one of the main information channels in interpersonal communication. Then, it is natural that research devoted to facial emotion has been increasing attention over the last years with applications not only in cognitive sciences but also in artificial intelligence.

The recognition of people and the identification of their corresponding emotions allow better interaction with computers, systems, and environments. Emotion recognition plays an important role in the prediction of social behavior [77], in emotional intelligence as an ability [78], and the experience of empathy [38].

Biometric patterns are categorized in physiological characteristics and behavioral characteristics. Whereas a biometric system based on physiological characteristics is more reliable, behavioral characteristics may be easier to integrate within certain specific applications. Examples of physiological characteristics include fingerprint, face recognition, palmprint, hand geometry, iris recognition, and retina. Behavioral characteristics are related to the pattern of behavior of a person including gait and voice.

This chapter addresses aspects of human emotion from a biometric point of view for the classification of emotion tasks. Specifically, emotions will be seen as a hidden biometric related to physiological and behavioral characteristics. Section 2.1 focuses on explaining the biometric foundations of emotional state and the motivation of its use against other techniques. The three principal approaches for modeling emotions we use will be explained. The idea of why topology is a useful tool to model emotion will be explained in Section 2.2. Finally, the state-of-the-art of the main approaches for emotion recognition existing in the literature is detailed in Section 2.3.

2.1 Emotions

Emotions are a vital part of humans, playing a valuable role in how a person perceives and understands the environment. Emotions can be seen as a kind of information to guide us to interact with the world. They can restrict or expand our behavior according to the situation. As emotions are so important, physiologists have formed several theories about how the emotions are generated and how important is the information they contain. Then, researchers have studied emotions and their theoretical development. The ever-growing theory raises several important questions that researchers must address as they bring emotion to their field [47]. For example:

1. Is it the physiological or the cognitive aspect of an emotional experience that primarily determine which emotion is being experienced?
2. Are emotions culturally specific or shared across cultures?
3. Are either emotions themselves or the causes that elicit them?

To answer these questions, we can say that our emotional states are combinations of **physiological arousal, psychological appraisal, and subjective experiences**. We defined them as the components of one emotion. Two people who faced the same

conjuncture could have a different emotion. Over time, different theories of emotion have been developed to explain how the various components of emotion interact with each other.

In the last decade, methods and algorithms have increased to facilitate emotion analysis; starting with manual methods using questionnaires made mostly by psychologists, continuing with complex methods involving computational algorithms. At present, emotion recognition through computers has a lot of application and it has been constantly developed by researchers.

2.1.1 Emotion modeling

Summing up, researchers have distinguished three principal approaches for modeling emotions [46]:

- Categorical approach: Based on the idea that only a small number of emotions exist, divided into six basic emotions: happiness, sadness, anger, fear, surprise, and disgust [48].
- Dimensional approach: Based on the idea that emotional states are not independent. On the contrary, they are related in a systematic way [48], covering the variability in three dimensions:
  1. Valence: how positive or negative emotion is.
  2. Arousal: how excited or apathetic an emotion is.
  3. Dominance:the degree of power.
- Appraisal approach: Based on the appraisal theory [25], it emphasizes the distinct components of emotions, and is often denominated componential view. This theory assumes that one emotion contains different emotional components including the subjective experience of the emotion itself, and it is given by a situational context. The occurrence of such emotional components can vary across different situations. For example, emotion based on a person's experience, opportunities for action goals, motivation, feelings, and reactions.

Emotions can be represented with a limited number of independent affective dimensions, modeled spatially in a circle, being arousal and valence their main characteristic features. See Figure 2.1 where the eight emotions considered in this thesis are placed regarding their arousal and valence. As we can see in Figure 2.1, the arousal

dimension is related to activity in both our mind and body, indicating the intensity of feeling along a single dimension ranging from sleep to frantic excitement. It links to attributes such as stimulated-relaxed, excited-calm, and wide awake-sleepy to define arousal [6]. The organs are stimulated with perception and the influence of the stimulation is related with arousal dimension. For example, when people are stimulated, they become aroused. Thus, it indicates how calming, soothing or exciting this stimulus is. The valence is usually represented in the horizontal axis and indicates the degree of pleasantness. It also indicates the emotional value that is associated with a certain stimulus and it is used to categorize emotions. It refers to the positive and negative character of emotions or some of its aspects.

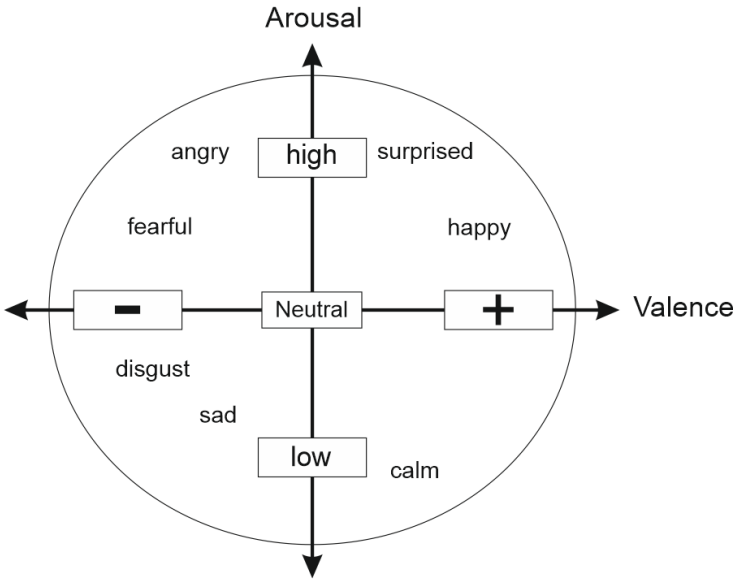

Figure 2.1: The eight emotions considered in this thesis placed spatially depending on their arousal and valence.

2.2 Topology to model emotions

Data analysis technologies including machine learning are based on statistical analysis that, until nowadays, has usually been the main technique of any research. Statistical analysis makes assumptions such as the data follows a normal distribution, so it is known that expected performance cannot be obtained if the data does not follow well-known distributions.

Topological Data Analysis (TDA) has emerged as an important approach for characterizing the behavior of datasets using techniques from topology. Tools from TDA, specifically persistent homology, allow assigning shape descriptors to large and noisy data across a range of spatial scales. Unlike deep learning, this technique does not rely on any label data training. TDA focuses on measuring topological summaries (e.g., connected components, loops, and voids) of the resulting patterns obtained from the data analyzed.

Learning high-level features from emotional utterances and creating a hierarchical representation of the signal is our main purpose. To address the problem, we will use TDA as a data analysis method that can capture in detail features and information by focusing on the shape of the data without using any statistical technique.

2.3 State-of-the-art approaches for emotion recognition

Emotion recognition has been an important topic for researchers where several machine learning techniques have been used. The three main theories that can be distinguished are discrete theory for categorical models, dimensional theory for dimensional models, and appraisal theory.

Discrete models describe an emotional state as discrete labels using basic emotions such as: sad and happy. In this way, complex affective emotions cannot be expressed. Dimensional models consider an emotional state as a point in a continuous space modeling more complicated emotions. Typically, an emotional state is covered by three dimensions: arousal (level of affective activation), valence (pleasure), and dominance (a measure of power or control), these dimensions describe many emotional states. Appraisal theory [101] focuses on detailing the mental processes underlying the incitement of emotions. An emotional state is interpreted as a control set of stimulus evaluations.

While emotions can be expressed in different ways, automatic recognition has mainly focused on facial expressions and speech [102]. There are a few articles about body gestures and posture on emotional recognition summarized in [91]. Nevertheless, the most intuitive model is to detect facial expressions and analyze the emotion later. This is very useful in the case of a short distance from the person.

Recognizing emotion from facial expressions has several advantages (+) and disadvantages (-) such as:

+ There are many datasets available for facial expressions.
+ The tools created for other purposes on facial recognition can be used.
+ It is considered the natural way to identify the emotional states.
- It is determinant to have a high quality of the video and a good segmentation.
- Actors can fake the movements involved in facial emotions.
- The information of the context is not provided, thus sometimes the outputs are misleading.

Data is sequences of images together with audio signals that can represent the temporal characteristics of emotion. Usually, in the process, it is necessary to enhance the quality of the image, for example, by changing the contrast and brightness or removing the noise.

In the literature, there are several algorithms to solve illumination problems. Besides, it is required to extract the face from the video to apply later a detection face algorithm, where the most famous is the Viola-Jones algorithm. One relevant analysis of this approach is found in [130]. Another step is to extract features from the video: geometric features, appearance features or a mix of geometric and appearance features are the principal ones.

Geometric features focus on the relationship between facial components. Features based on the position and angle of landmark facial points [39] will be used in this work. The appearance features are extracted from the global face region or from the analysis of regions containing different types of information [118]. Principal component analysis, Gavor-wavelets features, and local binary pattern histogram are the principal methodologies implemented in time real for this approach. The hybrid features bring better results in certain cases; the strength of both approaches generates an approach with fewer weaknesses [56].

Another approach is to recognize emotions using text whose solution is useful in different fields as data mining, information filtering systems, human-computer interaction, and psychology. As we see in [2], the solution to this problem is based on identifying keywords and assigning emotion to a text selected from a set of predefined emotion labels. Emotion recognition in the text is one of the difficult tasks in natural language processing and is very important as the principal medium of human-computer interactions through emails, chats, messages, forums, blogs, and other social

platforms. There are different levels of complexity. For example, one emotion can be expressed through the meaning of words and their relations or could be expressed by metaphors, irony, or sarcasm.

Another important branch focuses on the study of some parameters from the audio signal, such as a voice tone or prosody, highlighting the verbal communication. The features extracted from audio signals are used as input to emotion classification algorithms. Recall that a speech signal is the principal way of communication between humans. There exist efficient methods based on speech for machines to recognize human voices. However, we are still far from having a natural interaction between humans and machines because understanding the emotional state of the speaker is still an ongoing goal. This fact has motivated a recent research field namely speech emotion recognition. It is very useful for application webs and computer tutorials where the response of these systems to the user depends on the detected emotion [109], such as a diagnostic tool for the therapist [34] used in call center applications and mobile communication [75].

Speech emotion recognition task is very challenging because it is not clear which features are more distinguishable between emotions. There also exist other problems like variability, introduced by the existence of different sentences, speakers, speaking styles, and rates. Another issue is how a certain emotion is expressed. Generally, it depends on the speaker, his culture, and the environment or if he/she is speaking in his/her mother tongue. Although there are few works about multi-lingual classification [53], most work has focused on mono-lingual classification assuming there is no cultural difference among speakers.

Emotion does not have a common theoretical definition, people know the emotion when they feel them. Nevertheless, researchers were able to study and define different aspects of emotion. This approach has a variety of applications, and it can be redirected to the voice recognition field, call center, or customer services.

The approach to design a system for emotion recognition based on speech primarily includes two phases known as the feature extraction phase and features classification phase. The election of features for speech representation is needed to design an appropriate classification scheme and to choice, a useful speech database for evaluating the algorithm developed.

There are many reviews on speech emotion recognition such as [125, 32] surveying the speech features and the classification techniques used in this task. In the works of this area, it is common to divide the speech signal into frames, within each frame

the signal is considered stationary. Later, local or global features are extracted from each frame. Local features, as pitch and energy, are extracted from each frame. Global features are computed as statistics from all speech features obtained.

There are a lot of different opinions in the literature about which local or global features are more suitable for emotion recognition. Top papers addressing global features [114, 54] agreed that they are superior to local features in terms of classification accuracy and performance. However, researchers have asserted that global features are efficient only to distinguish between the emotion of high arousal versus lower arousal, failing on the classification of emotions with similar arousal such as, for example, anger versus joy. Another disadvantage is given by the loss of temporal information on the signal. Ayadi et. al. [32] exposed the idea that it is unreliable to use complex classifiers such as the hidden Markov model and the support vector machine with global features since the number of training vectors may not be sufficient for reliably estimating model parameters.

A third approach for feature extraction is based on segmentation speech signals and later computation of a feature vector for each phoneme segmented, observing the variation in the spectral shapes of the same phoneme under different emotions [71]. The poor performance of phoneme segmentation algorithms is a problem in this approach. An alternative method is to analyze the voiced segment (caused by vibrations of the vocal cord and are oscillatory) rather than each phoneme since it is easier to implement and feasible.

In the literature, many speech features have been applied. They can be grouped into four categories:

1. Continuous features: pitch, energy, formants.
2. Qualitative features: voice quality, harsh, tense, breathy.
3. Spectral features: ordinary linear predictor coefficients, short-time coherence method, least-squares modified Yule-Walker equations and others.
4. Features based on the Teager energy operator.

On the categorical approach, the states are limited to a fixed number and could be hard to focus on a complex emotional state. Nevertheless, these types of emotions are included in the dimensional approach. Some researchers address the pros and cons of each feature but, until now, no one can identify which category is the best one. In the last years, speech signal processing has been engaged by deep learning to a significant degree achieving competitive results in several applications. For example,

Zhao in [134] constructed two convolutional neural networks and long short-term memory (CNN LSTM) networks: one 1D CNN LSTM network and one 2D CNN LSTM network, to learn local and global emotion-related features from speech and "logmel" spectrogram, respectively.

Most of the existing approaches are designed for specific databases, that make the full solution to this problem still tough. While the system is trained on a particular database, it continues facing different issues as ethnicity, appearance, culture, sex, age contextual meaning of sentences, and noise of background in the signal. According to this, the algorithm implemented in [72] does not work well when dealing with the natural environment. To recognize emotional states in a natural environment is challenging due sometimes we have to manage large volumes of unsegmented, non-prototypical, and non-preselected data. A model that allows us to work with different kinds of information into it is essential.

Then, to increase the performance of emotion recognition, there are lines of research mixing multimodal approaches in that audio feature with facial features and body gestures that overcome the performance of the mono-modal approaches.

This problem has been approached, analyzing audio features such as spectral or voice quality and video features including local phase quantization. This problem has been approached, analyzing audio features such as spectral or voice quality and video features including local phase quantization. For example, Nicole et al. [90] developed a system that performs emotion recognition based on multi-modal features as facial appearance, head movements of speakers, and spectral features from audio combined via a regression classifier. Deep learning has been used in audio-visual emotion recognition to improve the linear relationship between the features by capturing complex non-linear feature interactions in multimodal data [64].

According to what is discussed in this chapter, Topology Data Analysis opens its door to set the problem of classifying emotional state. It will be necessary to find a topological signature for emotions.

3 An overview of tools from TDA

We will assume a basic knowledge of computational topology. In [50] and [30], readers interested in the topic can address it in greater depth. In this chapter, we present some results that are not common in standard courses.

Topological Data Analysis (TDA) consists of measuring topological features of shapes and functions. The tools used for such measuring are usually borrowed from the field of computational topology. Homology is one of such tools, and for TDA purposes, it is computed on a filtered space or, equivalently, on a sequence of sub-level sets of a real-valued function on this space.

3.1 Simplicial complexes

To introduce objects in the computers it is common to use a scanner 3D which generates a point cloud of the edge surface of objects and later obtains a triangulation. This way of representing objects on computers is known as simplicial complexes and their elements, vertices, edges and triangles, as simplices. A d-simplex is a geometric object with $(d+1)$ vertices ("corners") that lives in a d-dimensional space.

Definition 3.1 (d-simplex). *A d-simplex is the convex hull of $d+1$ affinely independents points $S = \{v_0, v_1, \cdots, v_d\}$. The points of S are the vertices of the simplex:*

$$[v_0, v_1, \cdots, v_d] = \left\{ u \mid \sum_{i=0}^{d} \lambda_i v_i = u, \sum_{i=0}^{d} \lambda_i = 1,\ \lambda_i \geq 0\ \forall i \right\}.$$

All possible convex combinations $\sum_{i=0}^{d} \lambda_i v_i$ are in the convex hull of the set $S = \{v_0, v_1, \cdots, v_d\}$. For example, in Figure 3.1, the point P is a convex combination of the three points V_1, V_2, V_3, i.e., $P = \lambda_1 V_1 + \lambda_2 V_2 + \lambda_3 V_3$, where $\sum \lambda_i = 1$ and $\lambda_i \geq$

$0, \forall i$. All points that are in the shaded region, bounded by the edges of the triangle, are convex combinations of $\{V_1, V_2, V_3\}$ and constitute the convex hull of $\{V_1, V_2, V_3\}$ which coincides with a 2-simplex in $\mathbb{R}^2$. On the other hand, the point Q is not part of the convex hull.

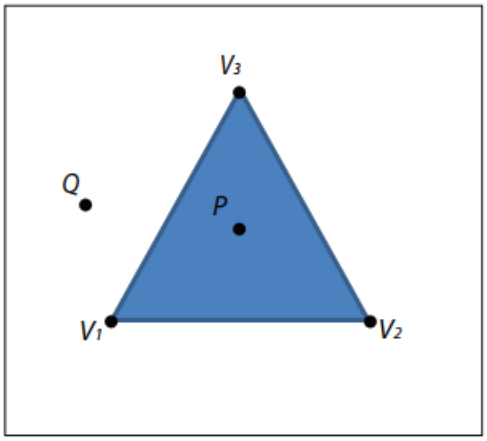

Figure 3.1: The point P is a convex combination of the three points V_1, V_2, V_3 and the point Q is not a convex combination of these points.

The first low dimensional simplices have their own names: 0-simplex, 1-simplex, 2-simplex and 3-simplex are also called vertex, edge, triangle and tetrahedron, respectively. Figure 3.2 shows them. The idea is easy: one vertex generates a point, two vertices generate a segment by connecting the two points, three vertices generate a triangle by connecting every pair of points with segments and filling the space between, and so on. Notice how $d + 1$ vertices are needed to generate an object of dimension d. This way, all possibles convex combinations of points in a given set $S = \{v_0, v_1, \cdots, v_{d+1}\}$ are in the convex hull of S. For example, in Figure 3.1, the point P is a convex combination of the three points V_1, V_2, V_3, i.e., $P = \lambda_1 V_1 + \lambda_2 V_2 + \lambda_3 V_3$, where $\sum \lambda_i = 1$ and $\lambda_i \geq 0, \forall i$. All points that are in the shaded region, bounded by the edges of the triangle, are convex combinations of $\{V_1, V_2, V_3\}$ and constitute the convex hull of $\{V_1, V_2, V_3\}$ which coincides with a 2-simplex in $\mathbb{R}^2$. On the other hand, the point Q is not part of the convex hull.

Definition 3.2 (face). *Any non-empty subset T of the point set $S = \{v_0, v_1, \cdots, v_d\}$ spans a simplex $\sigma_T \subseteq \sigma$ called a face of σ and denoted $\sigma_T \leq \sigma$. If T is a proper subset of S, then σ_T is called a proper face of σ. The boundary of σ, denoted as $\partial\sigma$, is the union of all proper faces of σ.*

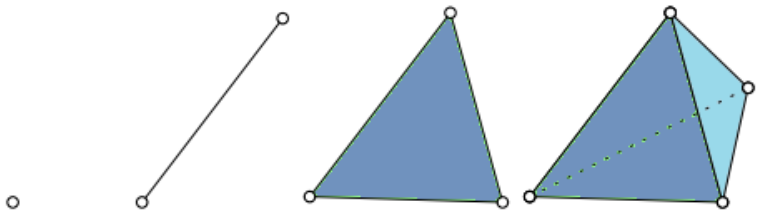

Figure 3.2: From left to right: vertex, edge, triangle, and tetrahedron.

Then, a simplicial complex is defined as follows.

Definition 3.3 (simplicial complex). *A simplicial complex K is a set of simplices such that:*

- $\sigma \in K, \tau \leq \sigma \rightarrow \tau \in K$.
- $\sigma, \sigma' \in K \rightarrow \sigma \cap \sigma' \leq \sigma, \sigma'$.

The dimension of the simplicial complex is given by the largest dimension of its simplices according to $dim(K) = \max\{dim(\sigma) \mid \sigma \in K\}$.

Definition 3.4 (subcomplex). *A subcomplex of K is a subset $K' \subset K$ which itself is still a simplicial complex.*

Finally, the n-skeleton of a simplicial complex $K^{(n)}$ is the subcomplex of K consisting of the restriction of the latter to its simplices of degree at most n, i.e., $K^{(n)} = \{\sigma \in K | dim(\sigma) \leq n\}$.

3.2 From point cloud to simplicial complexes

In TDA, it is assumed that the data are sampled from an underlying space $\mathbb{X}$, and the aim is to recover the topology of $\mathbb{X}$. Generally, the process follows these steps:

1. Find an approximation of $\mathbb{X}$ using a combinatorial structure such as simplicial complexes.
2. Use techniques from algebraic topology such as persistent homology to compute topological invariant of such structure.

There are several methods to complete the first step of the analysis. We can divide them into geometric and algebraic. The Czech complex and the Vietoris-Rips complex [116] are the most common algebraic methods. The Czech complex is computationally expensive in high dimensional analysis, being Vietoris-Rips a more efficient approximation technique.

Other efficient methods have been introduced in order to reduce the computational cost such as alpha complexes [31] and flow complexes [42]. They have the advantage of being fast and relatively small, but unfortunately, they depend on the Delaunay complexes [124].

Although the Delaunay triangulation produces a result in shape more interesting that the convex hull, it exhibits a dense partition of the space and, in particular, to recover the topological information from the shape as connected components, holes, and voids are very difficult tasks. Topics related to the two most used methods built on random points are addressed below, where points are embedded in an Euclidean space $\mathbb{R}^d$.

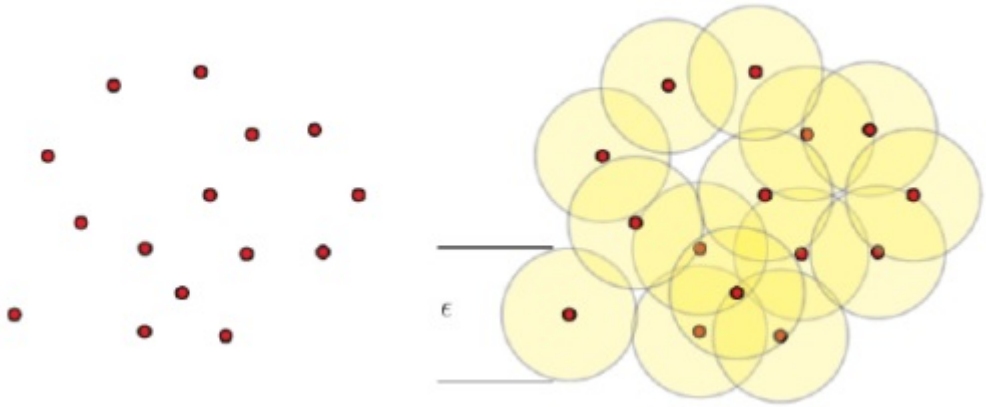

Figure 3.3: Left: a point cloud. Right: the union of balls centered at the points in the point cloud.

The main problem when using tools from simplicial homology to study a dataset $\mathbb{X} = \{x_i\}_{i=1}^m \subset \mathbb{R}^n$ is that there is not a simple structure to describe the data. Finding a simplicial complex from $\mathbb{X}$ could be difficult. The first strategy is to consider the homology of the spaces $\mathbb{X}_\epsilon = \cup_{i=1}^m B(x_i, \epsilon)$, where the ball of radius ϵ is computed around each point of $\mathbb{X}$. Then, the union of balls, $\mathbb{X}_\epsilon$, constitutes a good combinatorial

descriptor (see Figure 3.3). The spaces $\mathbb{X}_\epsilon$ with $\epsilon > 0$ are known as level sets which are induced by a function, in this case, the one defined by the enlargement of the balls. The level sets of a function in a finite point cloud $\mathbb{X}$ are also finite (although the parameter ϵ is continuous). Then, a filtration is a sequence of level sets which is a sequence of simplicial complexes satisfying that $K_1 \subset K_2 \subset \cdots \subset K_r$. An example of a filtration is shown in Figure 3.4.

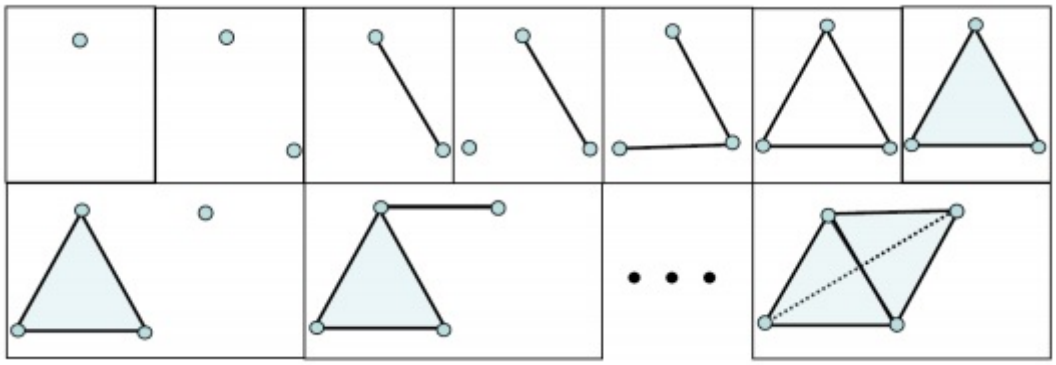

Figure 3.4: Example of a filtration.

3.2.1 Algebraic methods to compute filtrations

Two algebraic approaches to compute filtrations from point clouds are Czech complexes and Vietoris-Rips complexes.

Czech complexes

The Czech complex generated by a set of points $\mathbb{X}$ is a simplicial complex formed by vertices, edges, triangles, and faces of higher dimension. Although the general definition is quite wide, most articles reviewed work on a special case developed using the intersection of enough Euclidean balls.

Definition 3.5 (Czech complex). *Let $\mathbb{X} = \{x_1, x_2, \cdots, x_n\}$ be a collection of points in $\mathbb{R}^d$ and let $\epsilon > 0$. The Czech complex is defined as follows:*

1. *Its 0-simplices are the points in $\mathbb{X}$.*
2. *A k-simplex $[x_{i_0}, \cdots, x_{i_k}]$ is in $\hat{C}_\epsilon(\mathbb{X})$ if $\cap_{n=0}^{k} B_\epsilon(x_{i_n}) \neq \emptyset$.*

A paradigm on TDA is to create an estimation $\hat{U}_\epsilon(\mathbb{X})$ of any underlying submanifold, $\mathcal{M} \subset \mathbb{R}^d$, from which $\mathbb{X}$ is the sample and then consider its homology, through what is known as persistence homology or through Betti numbers as it is addressed (see [95]). Readers interested in homology theory may refer to [52, 50]. An important result in this area is the Nerve Lemma [9], which establishes that the Czech complex and the neighborhood set $\hat{U}_\epsilon(\mathbb{X})$ are homotopically equivalent and, in particular, they have the same homology groups. However, since the definition of the Czech Complex is essentially combinatorial, the Czech complex is computationally more accessible and therefore more widely used in applications than the neighborhood set. The interest in working with the Czech complexes is since it is primarily a high-dimensional complex analogous to a geometric graph. For an extensive study on geometric random graphs, we can cite [98]. The Czech topology is closely related to the topology of the alpha complex. However, when the dimension is greater than three its calculation becomes impractical.

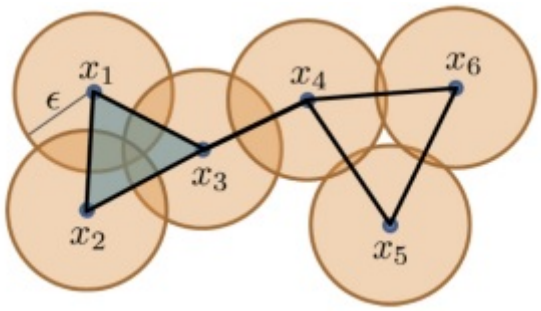

Figure 3.5: The Czech complex $\hat{C}_\epsilon(\mathbb{X})$ for $\mathbb{X} = \{x_1, x_2, \cdots, x_6\}$ and $\epsilon > 0$. The complex contains six vertices, two edges and one triangle.

Many times, there is not feasible to compute the Czech complex and the alpha complex can be applied only on dimension three or less. Another method use to compute filtrations, closely related to the previous one and easier to compute, is explained below.

Vietoris-Rips complexes

The Vietoris-Rips complex was introduced by Vietoris in [126] to extend the simplicial homology to a theory of homology of spaces in more general metrics. Although they are generally not as fast as alpha complexes at low dimensions, its calculation can be efficient in high dimensions.

Definition 3.6 (Vietoris-Rips complex). *Let $\mathbb{X} = \{x_1, x_2, \cdots, x_n\}$ be a collection of points in $\mathbb{R}^d$ and let $\epsilon > 0$. The complex $\hat{R}_\epsilon(\mathbb{X})$ is defined as:*

$$\sigma = \{x_1, x_2, \cdots, x_k\} \in \hat{R}_\epsilon(\mathbb{X}) \longleftrightarrow ||x_i - x_j|| \leq \epsilon \quad \forall i, j \in \{1, \cdots, k\}.$$

The definitions of the Czech and Vietoris-Rips complexes are not limited only to the case of the Euclidean space, they can be defined for a set of points in any metric space [17]. They satisfy the following property that plays an important role in TDA.

Lemma 3.1 ([26]). Let $\mathbb{X}$ be a finite set of points in $\mathbb{R}^d$ and let $\epsilon \geq 0$. Then there is a chain of inclusion maps:

$$\hat{R}_\epsilon(\mathbb{X}) \to \hat{C}_{\sqrt{2}\epsilon}(\mathbb{X}) \to \hat{R}_{\sqrt{2}\epsilon}(\mathbb{X}).$$

This means that a topological property that persists under the inclusion $\hat{R}_\epsilon(\mathbb{X}) \to \hat{R}_{\epsilon'}(\mathbb{X})$ with $\epsilon' \geq \sqrt{2}\epsilon$ is a topological feature of $\hat{C}_{\epsilon'}(\mathbb{X})$. The main idea is that the information about the topological features which persist under the previous inclusion reveal more information than if the same information is considered separately.

In [40], from a computational point of view, it is analyzed that the Vietoris-Rips complex is computationally less expensive than the corresponding Czech complex, despite having more simplices.

An unsatisfactory aspect of the previous results is the dependence on a priory knowledge of ϵ. One way to make a good choice of ϵ is to consider homological invariant multi-scale that encode changes in the shape of homology when ϵ changes. In [93], it is presented how to select the parameter ϵ; for enough small ϵ, the complex is a discrete set and for higher ϵ, the complex has a high dimension.

3.2.2 Geometric methods

A persistent problem in the methods presented above is that the simplices can have very high dimensions. Then, we address below some complexes exposed in [10, 70] that arise from computational geometry techniques to approach the problem.

Delaunay complex

To avoid the computational problems of Czech complexes and Vietoris-Rips complexes, it becomes necessary to limit the number of simplices in high dimensions. Delaunay complexes yield geometric tools to achieve this task and, nowadays, most of the simplicial complexes used are based on variations of Delaunay complexes.

Definition 3.7 (Voronoi diagram and Delaunay complex). *Let S be a set of points in $\mathbb{R}^d$. Define V_s as the set of points of $\mathbb{R}^d$ that is closest to $s \in S$ than to any of the points of S. That is, for $s \in S$, let us define:*

$$V_s = \{x \in \mathbb{R}^d \mid d(x,s) \leq d(x,s') \; \forall s' \in S\}.$$

The collection of the sets V_s is a cover for $\mathbb{R}^d$ and it is called the Voronoi decomposition of $\mathbb{R}^d$ concerning S. The nerve of this cover is called the Delaunay complex of S, denoted by $Del(S;\mathbb{R}^d)$.

The construction of this complex is costly in high dimensions, although efficient algorithms for the computation of Delaunay complexes for $d = 2$ and $d = 3$ have been developed. See [120] for more details on Voronoi diagrams and Delaunay complexes.

3.3 Homology

The n-dimensional homology group of a topological space represents the n-dimensional holes of the space. Intuitively, a 0-dimensional hole is a connected component, a 1-dimensional hole is a tunnel and a 2-dimensional hole is a cavity. Higher-dimensional homology groups do not have so clear intuition on the space but classical results ensure that n-dimensional homology groups are topological invariant, that is, they are invariant under homeomorphisms[1].

For each integer $n > 0$, the n-dimensional homology group of a topological space can be computed as follows. Suppose we have a topological space structured as a cell complex, being an n-dimensional cell a topological space homeomorphic to an n-dimensional ball. A 0-dimensional ball is a point(vertex), a 1-dimensional curve (edge). A 2-dimensional ball is homeomorphic to a disk and so on.

A cell complex K is then a collection of cells constructed inductively:

[1] A homeomorphism is a bicontinuous and bijective function between two topological spaces.

(1) The 0-skeleton $K^{(0)}$ of K (i.e., the set of 0-cells) are a set of points in an ambient d-dimensional space $\mathbb{R}^d$.
(2) Inductively, form the n-skeleton $K^{(n)}$ of K from the $(n-1)$-skeleton $K^{(n-1)}$ by attaching n-cells via homeomorphisms.

The boundary of an n-cell σ can be informally defined as the $(n-1)$-cells in the $(n-1)$-skeleton $K^{(n-1)}$ of K used to attach the n-cell σ. This way, if two cells intersect, they intersect at their boundaries. If a cell is in K, then the cells in its boundary are in K, where, for example, the boundary of an edge is its two endpoints (vertices). By abuse of language, we will say that a vertex is in the boundary of an n-cell if the vertex was one of the cells in $K^{(n-1)}$ needed to attach the n-cell σ. A maximal cell of K is a cell that is not in the boundary of any other cell of K. The dimension of K is the dimension of the cell of a higher dimension in K. From now on, we will assume that K has a finite number of cells.

We can formally sum cells of same dimension n to obtain the n-dimensional chain group $C_n(K)$, for each n. Moreover, the boundary operator is extended to a linear map ∂_n from $C_n(K)$ to $C_{n-1}(K)$ in the obvious way: the boundary of an edge is the alternative sum of its endpoints, and so on. Since the boundary of the boundary of a cell is always zero, then the image $B_n(K)$ of ∂_{n+1} is a subgroup of the kernel $Z_n(K)$ of ∂_n and then, the n-dimensional homology group of K is the quotient group $H_n(K) = B_n(K)/Z_n(K)$. An element α of $H_n(K)$ is called an n-dimensional homology class of K.

The rank β_p of $H_p(\mathbb{X})$ is called the p-th Betti number where, in the case of the three first dimensions, an intuitive meaning exists since β_0 is the number of connected components, β_1 the number of holes and β_2 the number of voids [33].

Simplicial complexes can be seen as particular cases of cell complexes. Simplicial homology consists of the homology groups of simplicial complexes and it is used to summarise the global connectivity of a topological space $\mathbb{X}$ decomposed as a simplicial complex, associating it with a sequence of abelian groups.

3.4 Persistent homology

The concept of homology is not useful in practice due to its lack of discrimination. This is why Edelsbrunner et al., in [30], introduced a more discriminating tool based

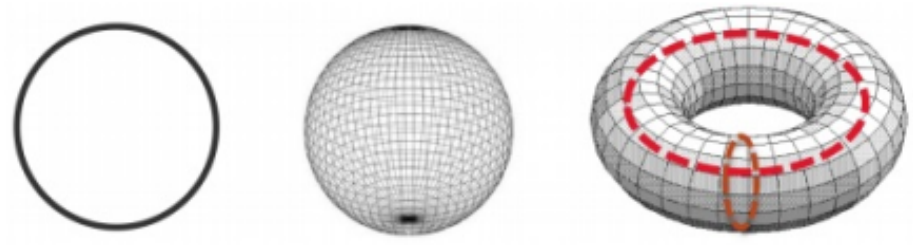

Figure 3.6: The circumference has one connected component and one hole ($\beta_0 = 1, \beta_1 = 1$). The sphere has one connected component and one void ($\beta_0 = 1, \beta_1 = 0, \beta_2 = 1$). The torus has one connected component, two holes and one void ($\beta_0 = 1, \beta_1 = 2, \beta_2 = 1$).

on homology, called persistent homology, together with an efficient algorithm for computing it. Later, Carlsson et al., in [135], reformulated and extended the initial definition.

Persistent homology applied to a filtration of a simplicial complex is in charge of keeping track of the moment i where a homology class is born and the moment j where the same class is died leading to topological descriptors as persistence diagrams or barcodes. Specifically, given a real-valued function $f : \mathbb{X} \to \mathbb{R}$ defined for a triangulable subspace of $\mathbb{R}^D$, persistence homology describes the variation in the topology of the lower level sets $f^{-1}(-\infty, t]$ when t increases from $-\infty$ to $+\infty$. For example considering the lower level sets $L_t = \{x \in \mathbb{X} : f(x) \leq t\}$, the index t means a scale parameter that leads the subspace filtration $\{L_t\}$ such that $L_t \subseteq L_s, \forall t \leq s$. The filtration leads a family $\{H(L_t) : t \in \mathbb{R}\}$ of homology groups and the inclusion $L_t \to L_s$ leads a family of homomorphisms $H(L_t) \to H(L_s)$.

Filtering functions

Given a cell complex K, persistent homology measures homology by filtration during a time, obtaining births and deaths of each homology class. Concretely, recall that a filtration is a finite increasing sequence of simplicial complexes:

$$\emptyset = K_0 \subset K_1 \subset K_2 \subset \cdots \subset K_n = K$$

Assuming that the vertices of a simplicial complex are points in the d-dimensional space $\mathbb{R}^d$, then a filtration can be derived from a real-valued map h on the set V of

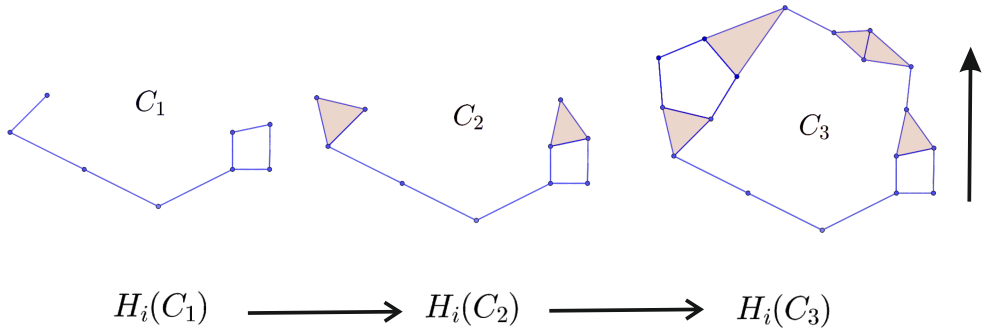

Figure 3.7: Top: Example of a filtration obtained using the height function on its vertices. Bottom: Associated 1-dimensional persistent homology.

vertices of K. In fact, there are several ways to obtain a filtration from h. For example, the one used in [44] for developing an emotion recognition method was the lower-star filtration. The one used in the first part of this work is different and defined as follows:
For each maximal simplex σ in K the h-value of σ is defined as:

$$h(\sigma) = \sum_{i=1}^{m} h(v_i)$$

assuming that $\{v_1, \dots, v_m\}$ are the vertices in the boundary of σ. Then, the filtration is:

$$\emptyset = K_h[0] \subset K_h[\ell_1] \subset K_h[\ell_2] \subset \cdots \subset K_h[\ell_n] = K.$$

where $K_h[\ell_i]$ is composed by the maximal simplices of K with h-value less or equal to ℓ_i and all the simplices in their boundary, $0 < \ell_1 < \ell_2 < \cdots < \ell_n$ and ℓ_n being large enough such that $h^{-1}(-\infty, \ell_n] = K$. See Figure 3.7.

Working over a field as the ground ring, persistent homology captures homology variations throughout a filtration. Assuming we have a filtration of the form:

$$\emptyset = K_0 \subset K_1 \subset K_2 \subset \cdots \subset K_n = K,$$

for each t and n in $\mathbb{Z}$, the n-dimensional homology group $H_n(K_t)$ is a vector space. For every $a \leq b$ and n, consider the linear maps $v_n^{a,b} : H_n(K_a) \to H_n(K_b)$ induced by the inclusion $K_a \hookrightarrow K_b$. The family of vector spaces $\left(H_n(K_t)\right)_{t\in\mathbb{Z}}$ together with the linear maps $\left(v_n^{a,b}\right)_{a\leq b}$ is called the n-dimensional persistent homology of the given filtration.

To compute the persistent homology of a cell complex, the notion of AT-model [45] is used in this work. We assume that the ground ring is $\mathbb{Z}_2$. See Algorithm 1 in page 50.

3.4.1 Persistence diagrams

Based on what was stated in the previous section, considering that the homology class α was born in $H_n(K_t)$ and died in $H_n(K_s)$, set that $b(\alpha) = t$ and $d(\alpha) = s$. Representing every class α by a point $(b(\alpha), d(\alpha))$ results a multi-set of points in $\mathbb{R}^2$ with the corresponding horizontal axis at the birth of the class and the vertical axis at death. The persistence of α is the difference $pers(\alpha) = d(\alpha) - b(\alpha)$, where in a general context, the set of points with infinity persistence corresponds to points of the form $(t, +\infty)$.

Definition 3.8 (persistence diagram). *A persistence diagram is a multiset of points in $\mathbb{R}^2$ with the diagonal*

$$\Delta = \{(t, s) \in \mathbb{R}^2 | t = s\},$$

where every point on the diagonal has infinite multiplicity.
Each point in the persistence diagram corresponds to a pair of birth and death time of a homology class. That is, if a point (t, s) belongs to a persistence diagram representing the persistent homology associated to a filtration, then there is a homology class α that was born in $H_n(K_t)$ and dies entering $H_n(K_s)$.
When the pairs $[t, s]$ represent intervals in the real plane then the set of such intervals is called persistence barcode.

Definition 3.9 (total persistence degree). *The total persistence of degree p of a persistence diagram d is defined as:*

$$Pers_p(d) = \sum_{x \in d} (pers(x))^p.$$

Note that the persistence diagram is included in the half-plane above the diagonal Δ since deaths always occur after births. In [16], it is shown how these diagrams are well defined for any metric space and in particular for any compact metric space. The features of higher persistence are represented by the points furthest from the diagonal while nearby points to the diagonal may be interpreted as topological noise.

Persistence diagrams are stable since a small change in the input function produces a small change in the diagram. There are different choices of metrics in the

space of persistence diagrams, analogous to the variety of metrics in the space of functions. In a general way, a metric distance is interpreted as a distance L^p in the function space of a discrete space. A natural family of metrics is discussed in [121].

Approaches to study the persistence diagrams from a probabilistic and statistical point of view are presented in [122, 103, 89, 18]. To use the persistence diagrams as a true statistic tool, natural questions about these summaries arise:

- Is it possible to define probability measures on these summaries?
- Can we establish relationships between the sampling distribution of the data and the distribution in the topological summary?
- Can we compute the mean and variance?

There is a variety of reasons for characterizing the statistical properties of diagrams. For example, given a point cloud S, it is advantageous to work with subsampling of the data which produce point clouds smaller and later to calculate the mean and variance of the persistence diagram set. In statistic terms, it consists of calculating a bootstrap estimation of a persistence diagram from data [18]. But these procedures require a good definition of the mean and variance. In the literature, some progress has been developed. In [84], Mileyko proved that the space of diagrams (Dgm_p, W_p) is a Polish space (with complete and separable metric), and therefore it is possible to define the Frechet mean. In particular, it has been proven that the Frechet measure of a finite set of persistence diagrams always exists but is not necessarily unique.

3.4.2 Distances between persistence diagrams

Let us define several distances between persistence diagrams for pairwise comparison between them.

Definition 3.10 (Wasserstein distance). *The Wasserstein distance between two persistence diagrams, d_1 and d_2 is defined as:*

$$W_p(d_1, d_2) = (inf_\gamma \sum_{x \in d_1} ||x - \gamma(x)||_\infty^p)^{\frac{1}{p}}$$

where γ is a set of all bijections between d_1 and d_2, it is not-empty due the diagonal.

Particular case: *For $p = \infty$, the bottleneck distance is defined as follows:*

$$B_\infty(d_1, d_2) = \inf_\gamma \sup_{t \in d_1} ||t - \gamma(t)||_\infty.$$

The empty persistence diagram is denoted by $d_\emptyset$, and is the diagram which includes only the diagonal; note that $Pers_p(d) = 2^p(W_p(d, d_\emptyset))^p$ for $t \in d$.

The metric W_p is used to define the following persistence diagram space [84]:

$$D_p = \{d \mid W_p(d, d_\emptyset) < \infty\} = \{d \mid Pers_p(d) < \infty\};\ p \geq 1.$$

The bottleneck distance is the most used metric to compare persistence diagrams due to its optimal properties and stability [29]. An intuitive idea of this distance is shown in Figure 3.8.

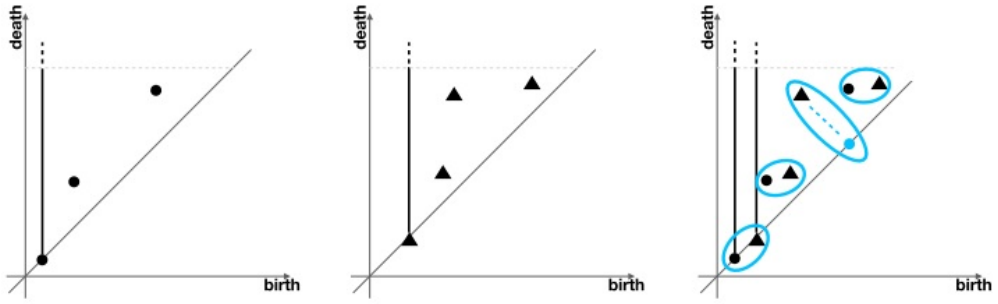

Figure 3.8: Part a) shows two persistence diagrams, part b) shows the match given by choosing the bijection which minimises the sum of the distances. Note that one corner-point represented as triangle is matched with its projection on the diagonal.

Now, let us describe several stability results on persistent homology, supporting the idea that an algorithm designed using persistent homology tools will produce "similar" outputs for "similar" inputs.

Theorem 3.1 ([22]). *Let $f, g : X \to \mathbb{R}$ be two tame[2] Lipschitz function on a metric space X whose triangulation grows polynomially with constant exponent $j \geq 1$. Then, there are constants $c > 1$ and $k \geq j$ such that the p-th Wasserstein distance between their corresponding persistence diagrams, denoted by d_f and d_g, satisfies:*

$$W_p(d_f, d_g) \leq c||f - g||_\infty^{1-k/p},\ \forall p \geq k.$$

[2]A function is tame if there is a finite number of elements in the set $\{H_m(f^{-1}(-\infty, a]\}$ and such set consists of homology groups whose ranks are finite, for $a \in \mathbb{R}$ and $m \geq 0$ being integer.

When $p = \infty$, the constant c is not longer necessary, obtaining the following most commonly used simplified version.

Corollary 3.1 ([30] p. 183). Let K be a simplicial complex, and let $f, g : K \to \mathbb{R}$ be two monotonic functions If d_f and d_g denote the corresponding persistence diagrams obtained from f and g, then:

$$B_\infty(d_f, d_g) \leq ||f - g||_\infty.$$

As a consequences of the previous theorem, the following theorem is guaranteed.

Theorem 3.2 ([15]). *Consider two finite metric spaces X and Y. Let d_f, d_g be the two persistence diagrams obtained, respectively, from Vietoris-Rips filtration. Then,*

$$B_\infty(d_f, d_g) \leq GH(X, Y)$$

where GH denotes the Gromov-Hausdorff distance.

Due to these results, we can assert that stability results are simpler when using the bottleneck distance than when using the Wasserstein distance. Given the above intuitive idea and the notation, the stability of the persistence diagram is defined as:

Theorem 3.3 (Stability Theorem [30]). *For a triangulable space X, two continuous and tame functions $f, g : X \to \mathbb{R}$, and any dimension $p \geq 0$, the bottleneck distance of the two p-dimensional persistence diagrams d_f and d_g is bounded by the distance between the functions:*

$$B_\infty(d_f, d_g) \leq ||f - g||_\infty$$

The proof of this result is very technical and uses commutative diagrams of vector spaces of homology classes. It can be consulted in [21].

3.4.3 Persistent entropy

Due to the number of applications based on TDA, there are numerous available software packages to calculate and represent the persistent homology in almost all coding languages. A nice study of the performance of software packages is available in [93].

Persistence diagrams, persistence barcodes, and, more recently, persistence landscapes constitute a compact way to represent the information obtained from persistent homology. Although these topological summaries are metric spaces used to

compare persistent homology of the dataset based on the techniques described previously, they do not work properly for statistical analysis. For example, they fail to have a unique mean.

Let us see how it is possible to summarize the information described by persistent homology as a number. The Shannon entropy [115] of a probability distribution obtained from persistent homology is a natural candidate to describe persistent homology as a number. It is known as persistent entropy. There are a lot of applications of this concept in pattern recognition of signals [83, 105], complex systems [8], biological images [3], and clustering [129].

The key to persistent entropy is based on the idea that significant topological attributes should have long lifetimes, and features with short times of life are considered noise. The concept of k-significant intervals was introduced in [43], where fixing $k > 0$, an interval $[a, b)$ means k-significant if $k < b - a$ and persistence barcodes associated with different filtration could be compared using the significant interval as a measure. Nevertheless, another form of comparison between persistence barcodes or diagrams can be obtained in terms of entropy, lending out special attention to intervals that persist to infinity. There are different ways to analyze the end of the filter denoted by (a, ∞). In this work, this term is denoted by $(a, n + 1)$, where n is a fixed big positive integer. This way, all points in the persistence diagram have finite coordinates. The set of persistence barcodes with intervals of finite length only will be denoted by $\mathcal{B}_F$.

Definition 3.11 (persistent entropy [20]). *Given a filter F and the corresponding persistence diagram $B = \{[a_j, b_j] : j \in J\} \in \mathcal{B}_F$, the persistent entropy of F is defined as:*

$$E(F) = -\sum_{j \in J} p_j \cdot \log(p_j) \tag{3.1}$$

where, $p_j = \frac{l_j}{L}$, $l_j = b_j - a_j$, and $L = \sum_{j \in J} l_j$.

Observe that Formula 3.1 can also be written as follows:

$$E(F) = log(L) - \frac{1}{L} \sum_{j \in J} l_j \cdot \log(l_j).$$

The persistent entropy has been implemented as a method in Gudhi library[3], scikit-TDA library[4] and Giotto library[5].

[3] https://github.com/GUDHI/gudhi-devel/tree/master/src/python/gudhi/representations/vectormethods.py
[4] https://github.com/scikit-tda/persim
[5] https://github.com/giotto-ai/giotto-tda/blob/master/giotto/diagrams/features.py

In previous sections, we have analyzed the stability of persistence barcodes and diagrams. Now, we are going to show under which conditions persistent entropy is stable, that is, uniformly continuous or, in other words, the noise produced by data has practically no effect in the computation of persistent entropy.

Theorem 3.4 (Stability of persistent entropy [4]). *Let $A, B \in \mathcal{B}_F$. Let us assume that $r_p(A, B) \leq \frac{1}{4}$. Then:*

$$|E(A) - E(B)| \leq 2r_p(A, B)(log(n_a + n_b) - log(2r_p(A, B))),$$

where $r_p(A, B)$ with $1 \leq p \leq \infty$ is the relative error defined as:

$$r_p(A, B) = \frac{2(n_p)^{1-\frac{1}{p}}}{L_{max}} d_p(A, B),$$

and n_p denotes the cardinality of the bijection where $d_p(A, B)$ is reached.

3.4.4 Time-varying systems

A time-varying system can be interpreted as a series of relevant geometric and topological events. Persistent homology is usually applied to static point clouds and shapes by supplying a topological description of the analyzed space. It provides a representation in which the features obtained are ordered by relevance. These topological features help to identify interesting patterns in the data clustering in time series and spatial data [99]. This way, persistent homology has been generalized by [23] to time-varying systems considering continuous representation, or introducing statistics, to evaluate the evolution in time of the analyzed system [122].

Definition 3.12 (time-varying system). *A time-varying system is a system (X, t) whose dynamics change over time. Its output response depends on the moment of observation as well as the moment of the input signal. Then, a time delay or time advance of input leads not only to an appropriate time shift in the output signal but also to changes in other parameters of the output signals [19].*

A time series is simply a series of data points ordered in time where the time is often the independent variable and, usually, the goal is to make a forecast for the future. Time series data-mining arises from the necessity to codify the natural capacity of humans to visualize the shape of data and extract all meaningful knowledge from that

shape. The principal time series related tasks include query by content, anomaly detection, motif discovery, prediction, clustering, classification, and segmentation supported by a strong theoretical framework. A summary of these terms can be consulted in [62].

Definition 3.13 (time series). *A time series T is an ordered sequence of n real-valued variables*

$$T = (t_1, \dots, t_n),\ t_i \in \mathbb{R}.$$

Definition 3.14 (Subsequence of a time series). *Given a time series $T = (t_1, \dots, t_n)$ of length n, a subsequence S of T is a series of length $m \leq n$ consisting of contiguous time instants from T:*

$$S = (t_k, t_{k+1}, \dots, t_{k+m-1})$$

with $1 \leq k \leq n - m + 1$. The set of all sub-sequences of length m from T is called S_T^m.

If the geometry of the shape that we are analyzing changes then its persistence diagram should be updated, encoding the significance of that variation in time with respect to the filter function applied.

The generalization of persistent homology to time-varying systems was introduced by Edelsbrunner et al. in [21]. The principal idea is the following:
If we have a function that changes continuously, according to the Stability Theorem, its persistence diagram also changes continuously. Then, it is necessary to understand the changes and observe how the points move, and analyze the patterns. One possibility to study these patterns is stacking up the persistence diagrams, analyzing the trace of each point as a curve in the space.

Specifically, Edelsbrunner et al. in [21] considered the homotopy $F(x,t) : \mathbb{X} \times [0,1] \to \mathbb{R}$, where $\mathbb{X}$ defines a based space, and $f_t(x) = F(x,t)$ denotes a frame at a given time-slice. Assuming every f_t is tame, a p-dimensional persistence diagram $\mathrm{Dgm}_p(f_t)$ for every t and p is defined and the relation among persistence diagrams is given by the stability theorem. The points off-diagonal in $\mathrm{Dgm}_p(f_t)$ moves in time, forming a curve with its trace, referred to as a *vine*.

For example, a homotopy of piecewise linear functions arises when the frame f_0 and f_1 of the function $F(x,t)$ are interpolated. If the vertices follow a straight-line homotopy $f_t(v) = \lambda f_1(v) + (1-\lambda) f_0(v)$, then the order of the simplices in the lower-star filtration is determined by function $f_t(\sigma) = \max_{v \in \sigma} f_t(v)$ [86]. TDA applied for time series is relatively new and fast-growing, with many existing applications in several

different domains. For example, the 1-homology groups offered via Takens' embedding in windows of a time series is computed by Khasawneh [63] in order to track the stability of signals. As a way to quantify the periodicity of time series, Seversky et al. [111] used the maximum persistence of homology groups. Other authors have used the features from persistent homology to cluster [99] and as inputs in convolutional neural networks for classification of time series [123]. Applying TDA on financial time series has converted an important issue nowadays [41].

This thesis suggests combining techniques from TDA as persistent homology and time series in order to characterize them from a topological point of view and to analyze which are the benefits of such representation and how to distinguish between relevant and noisy signals.

3.5 Persistence time series

Given a time-varying system (X, t), the vineyards encode how homological critical values change around time by a given filter function. It is considered by researchers powerful tools for describing time-varying systems. But, to interpret this tool is very difficult and until now there is no general technique to compare vineyards associated with two time-varying systems [7]. Let, $\mathbb{T}$ be a topological space, $f : \mathbb{T} \times I \to \mathbb{R}$ a tame function for every $t \in I$, and $t = \{t_0, t_1, \ldots, t_n\}$ a set of $n + 1$ spaced points. Then, a p-dimensional persistence diagram $Dgm_p(T, f)_i$ is associated with every $t_i \in t$ and the collection of these p-persistence diagrams is a time series $Dgm_{p,n}(T, f) = \{Dgm_p(T, f)_i\}_{i=0}^{n} \subset Dgm_\infty$ where (Dgm_∞, d_B) is the space of persistence diagram with the associated bottleneck distance.

In the literature, there exist many methods to evaluate the dissimilarity of two-time series. The dynamic time warping [110] is considered in this work because it is a powerful technique that can be applied across many different domains and it is useful to compare the similarity between two-time series with different length.

3.5.1 Dynamic time warping

Intuitively the dynamic time warping (DTW) algorithm compares the similarity or, in other words, calculates the distance between two arrays or time-series under certain restrictions. This method has been used to compare different speech patterns

in automatic speech recognition. Such technique will be used in this work with the same purpose but with a different workflow. Let two time-independent sequences $X := \{x_1, x_2, \dots, x_n\}$ of length $n \in \mathbb{N}$ and $Y := \{y_1, y_2, \dots, y_m\}$ of length $m \in \mathbb{N}$. These sequences may be discrete signal, in our case time-series or, more generally, feature sequences sampled at equidistant point in time. If we fix a feature space $\mathcal{F}$, to compare two different features $x, y \in \mathcal{F}$, a dissimilarity function or cost function is needed, denoted as $c : \mathcal{F}x\mathcal{F} \to \mathbb{R}$, such that:

- $c(x, y) \geq 0, \ \forall x, y \in \mathcal{F}$,
- $c(x, y) = 0$ if and only if $x = y$.

Note that the cost function is not necessarily a metric. Intuitively, if $c(x, y)$ is small, x and y are similar to each other, and otherwise, if $c(x, y)$ is large, it means high cost. Evaluating the local cost measure for each pair of element of both time series, the cost matrix $C \in \mathbb{R}^{nxm}$ can be obtained, whose entries are defined as $C_{ij} := c(x_i, y_j)$.

A warping path [7] $\xi = (\xi_1, \xi_2, \dots, \xi_l)$ is a sequence or pairs of indices $\xi_k = (i_k, j_k)$, where $i_k \in \{1, \dots, n\}$, $j_k \in \{1, \dots, m\}$ and $k \in \{1, \dots, l\}$ holding the next conditions:

1. The starting and ending point of two time series have to be aligned.
2. Given $\xi_k = (i_k, j_k)$, and $\xi_{k+1} = (i_{k+1}, j_{k+1})$, then $i_k \leq i_{k+1}$ and $j_k \leq j_{k+1}$ holds, means that the natural ordered induced by the time is preserved.
3. The size of the shifts in time to align the two time series is constrained (difference between two paths $\xi_{k+1} - \xi_k \in \{(0, 0), (1, 0), (0, 1), (1, 1)\}$)

Then, the total cost is defined by:

$$c_\xi(X, Y) := \sum_{k=1}^{l} c(x_{i_k}, y_{j_k}).$$

The optimal warping path will be the minimum total cost, then the DTW distance is defined as:

$$DTW(X, Y) := \min \left\{ c_\xi(X, Y) \mid \xi \text{ is a } (n, m)\text{-warping path} \right\}$$

where the minimum always exists because the set is finite.

If we consider persistence time series then the bottleneck distance will be used as a cost function.

Definition 3.15 (dynamic time warping (DTW)). *Let $D_{k,n}(X, f)$ and $D_{k,m}(Y, g)$ be two k-dimensional persistence time series. Then, the dynamic time warping is defined as:*

$$DTW(D_{k,n}(X, f), D_{k,m}(Y, g)) = \min\left\{d_{B_\xi}(D_{k,n}(X, f), D_{k,m}(Y, g)) \mid \xi \text{ is a } (n, m)\text{-warping path}\right\}.$$

The cost matrix will be:

$$A(n, m) = DTW(D_{k,n}(X, f), D_{k,m}(Y, g)).$$

Note that DTW is well defined even if there are several warping paths of the minimal total cost. Besides, DTW is symmetric since the local cost is symmetric.

In this work, our goal is to extract, from the time series, topological information not available in standard form. Takens' embedding theorem is used to extract the attractor of the series and then do the analysis on it, guaranteeing the preservation of topological properties of time series.

Takens' embedding theorem [119] presents conditions under which a discrete-time dynamical system can be reconstructed from scalar-valued partial measurements of internal states. In other words, it is possible to reconstruct a time series considering time delays. The approach consists of converting a time series $\{x_t,\ t = 1, 2, \dots, T\}$, into its phase space, that is a point cloud

$$\left\{\{x_i, x_{i+\tau}, \dots, x_{i+d_\tau}\},\ i = 1, 2, \dots, T - d_\tau\right\},$$

where τ is a delay parameter and d specifies the dimension of the point cloud.

Theorem 3.5 (Takens' embedding theorem [119]). *Let M be an m-dimensional compact manifold. For any pair $(\phi.y)$ with $\phi \in \mathrm{Dif}^2(M)$ and $y \in C^2(M\mathbb{R})$, then the map $\varphi(\phi, y)$ is an embedding.*

A series $\{x_0, x_1, \dots, x_{n-1}\}$ can be reconstructed in phase space as

$$x_n(m, \tau) = (x_n, x_{n+\tau}, \dots, x_{n-(m-1)\tau})$$

where m is the embedding dimension and τ the time delay. Thus, instead of analyzing the series along time, we will study its trajectory as a set of states in an m-dimensional Euclidean space.

In order to analyze the evolution of a system in time as a collection of observations, windowing is defined as a uniform partition of the composition according to its subdivision in bars.

3.5.2 Embedding delay parameter selection

Time-delay embedding is a uniform subsampling of the original time series data according to an embedding parameter τ. It has been considered primarily in the context of analyzing dynamical system [119], where a time-delay embedding of time series data was used to recover the underlying dynamics of a system. In order to perform the phase space reconstruction, it is very important to select an appropriate pair of embedding dimension m and τ. The precision of these parameters is directly related to the characteristics of the attractors in phase space reconstruction. Aiming to obtain these values, there are two different approaches to the literature. The first one is that τ and m are not correlated with each other; that is, both can be selected independently. Then, different criteria can be used for both parameters. For time delay τ there are three different approaches:

1. Multiple auto-correlation and non-bias multiple auto-correlation [58].
2. Series auto-correlation such as auto-correlation, mutual information [36], high-order correlations [1].
3. Phase space extension, e.g., fill factor [12], wavering product [11], average displacement [104] and others.

Another approach more practical consists of using that m and τ are closely related since the time series data is not infinitely long and it is difficult to avoid noise. A lot of experiments done by researchers indicate that m and τ are related with time window $t_w = (m-1)\tau$ for the reconstruction of the phase space. From a practical point of view, this is more reasonable and the combination algorithm for dimensional embedding and delay time have become an important line of analysis in the category of the chaotic time series analysis.

3.6 Machine learning background

Machine learning usually refers to changes in systems that perform tasks associated with artificial intelligence. Such changes might be either improvement to already per-

forming systems or initial synthesis of new systems. Such tasks involve recognition, classification, planning, prediction, and others. Learning is a spectrum very wide. The field of machine learning has been divided into several subfields focusing on different types of learning tasks. In order to provide an idea of this wide field, we summarize the four parameters existing along which learning paradigms can be classified [113].

1. **Supervised versus unsupervised**: Learning involves an interaction between learner and the environment, then we can divide it according to this kind of interaction. The first one is the difference between supervised and unsupervised learning. On supervised learning, we know the values of the function f for the n samples in the training set. Then, to assume that there is a hypothesis that matches the function for the elements of the training set involves that the hypothesis will be a good guess for f. In the unsupervised case, a training set of vectors without function values is provided. Then, the problem here is to partition the training set into subsets in some appropriate way. This setting is very useful in taxonomic problems in which it is desired to research ways to classify data into meaningful categories.
2. **Active versus passive**: An active learner interacts with the environment at training time for example performing experiments. On another side, the passive learner only observes the information provides by the environment.
3. **Online versus batch learning protocol**: The distinction between environments in which a learner has to respond online, along with her learning process and the environment in which the learner applies experience acquired after had the opportunity to process a large amount of data.
4. **"Helpfulness of the teacher"**: When a scientist learns about the nature of the data, the environment plays the role of the teacher. To model such learning scenarios, we assert that the training data (or learning experience) is generated by some random process. Then, learning happens when the learner's input is given by an adversarial called the teacher.

Nowadays, machine learning techniques are widely applied to solve classification problems. A classification technique will use a training dataset:

$$D = \left\{(\vec{v}_i, c_i) \mid \vec{v}_i \in \mathbb{R}^n,\ c_i \in \{0, \dots, k\},\ i \in \{1, \dots, m\}\right\}$$

where the values c_i are the different k possibles classes that can exists, $\vec{v}_i$ vectors of dimension n that we are going to use in the classification. Through this dataset, the classification algorithm will produce a classification model.

The different existing settings in machine learning offers very useful tools; one of these is the Support Vector Machine (SVM) paradigm for learning linear predictors in high dimensional feature spaces. The high dimensionality of the feature space establishes a challenge given by sample complexity and computational complexity.

A SVM is a supervised learning technique that construct a hyperplane, driven by a linear function

$$b + \sum_{i=1}^{m} \alpha_i \vec{v_i}^T \vec{v}_i,$$

or a set of them that can be used to classify data. When this data is not linearly separable, a kernel trick is applied. The space is mapped to higher dimensions using a kernel function,

$$k(\vec{v}, \vec{v'}) = \phi(\vec{v}) \cdot \phi(\vec{v}').$$

Therefore, a support vector machine just creates hyperplanes that work as decision boundaries for classification after applying a deformation of the dataset to get a linearly separable representation. Then, formally, a SVM within a kernel makes predictions using the following function:

$$f(\vec{v}) = b + \sum_{i=1}^{m} \alpha_i k(\vec{v}, \vec{v}_i).$$

SVM techniques were originally conceived as efficient methods for pattern recognition and classification. Researchers have used these settings in several linear and nonlinear applications such as speech recognition, image processing, array processing, communication systems. discriminant analysis, clustering, and many other applications.

3.6.1 Machine learning with KNN Algorithms

What are KNN algorithms?

K-Nearest-Neighbor (KNN) algorithms are supervised learning machine methods used in classification and regression task. It assumes that similar features exist in close proximity. In other words, similar things are near to each other. Its main goal is captures the idea of similarity, that can be defined as distance, proximity, or closeness. The idea is easy to use, and does not make assumptions about the data. Its accuracy

depends on the quality of the data, and in the algorithm should find the optimal number of nearest neighbors (K-value) [82].

How do KNN algorithms work?

- INPUT: $(x_1, y_1), \cdots, (x_n, y_n) \in \ XxY$ (having X as the input space and Y their class labels)
- Load the dataset.
- Select the optimal value K of the neighbors.
- Calculate $d_i = d(x, x_i);\ i = 1, 2, \cdots, n$, where d denotes the distance defined.
- Sort in ascending order the collection of distances and indices.
- Take the first K distances from this sorted list.
- Get the labels of the selected K entries.
- In case of regression, return the mean of the K labels, and in classification return the mode.
- OUTPUT: D_x^K assigning x to its most frequent class

Choosing the right value for K

To select the right K for the specific dataset in analysis, the KNN algorithm runs several times with different values of K, and selects the K that reduces the number of errors found, while keeping the ability of the algorithm to accurately make predictions with data than it has not seen before.

3.6.2 Performance metrics

After knowing how to implement a machine learning model and get outputs in form of a probability or a class, the first model is rarely the best one. Then, it is needed to evaluate the quality of our machine learning model in order to improve the model until it performs as best as it can. In the literature, different performance metrics have been used to evaluate these algorithms. Basically, in this work, we will focus on the classification problem. Thus, we will use classification performance metrics such as accuracy, the area under the curve, precision, etc.

When conducting an experiment, the metric is chosen to evaluate the model is very important. Choosing the metric involves how the performance of the algorithm

is measured and compared. Next, let us see the most used metrics and the metric chosen in this work.

1. **Confusion matrix**: It is a very intuitive metric and used to find the correctness and accuracy of the model. It is useful if we need to classify the data into two or more types of classes. This technique itself is not a performance measure, but almost all metrics are based on it. In this case, we will define confusion matrix as a table with two dimensions ("Actual" and "Predicted"), and sets of "classes" in both dimensions. Usually, each column of the matrix represents the instances of the predicted class by the model applied, while each row the instance of the actual class in other words the class observed. Then, diagonal elements indicate correct predictions, while the off-diagonals represent incorrect predictions.

		Actual	
		Positive (1)	Negative (0)
Predicted	Positive (1)	TP	FP
	Negative (0)	FN	TN

   The terms associated with the confusion matrix are:

   - True Positive (TP): the actual and predicted classes of the data point are true.
   - True Negative (TN): both classes are negative.
   - False Positive (TP): the model predicts incorrectly, the actual class is false and the prediction class is true.
   - False Negative (FN): the actual class of the data point is true, and the model predicts it is false.

   It depends on the problem and the context if we need to minimize the false negatives or positives.
2. **Accuracy in the classification problem**: it is considered as the percentage of well-classified data in a dataset:

$$\text{Accuracy} = \frac{m}{n}$$

   where m is the number of well-classified data and n is the size of the full dataset used in the test. Accuracy is a good measure when the target variable classes in the data are nearly balanced. The "target variable" is the variable whose values are to be modeled and predicted by other variables (analogous to the dependent variable in linear regression). It should never be used if the target variable in

the data belong to one class. Let see one example, to understand better these terms.
In our happy emotion detection task with 200 videos of people expressing emotions, but only 10 people express happiness. Let's say our model is very bad and predicts every video of emotions as no expressing happiness. In doing so, it has classified those 190 non-happy emotions videos correctly and 10 happiness video as Non-happy. Then even though the model is terrible at predicting happy emotion, the accuracy of such a bad model is also 95%.

3. **Precision**: It is a measure that evaluates how close a measurement comes to another measurement.
4. **Sensitivity**: It measures how often the test captures the cases correctly. For example, positive results in the test for a person who has the disease.
5. **Specificity**: It measures the ability to correctly generate a negative result. For example, if a person does not have the condition that is being tested for it.

3.6.3 Statistical tools

The correlation coefficient of two random variables is a measure of their linear dependence. One correlation coefficient largely known and applied is the Pearson correlation coefficient.

$$\rho(A, B) = \frac{\text{cov}(A, B)}{\sigma_A \sigma_B}$$

where $\text{cov}(A, B)$ is the covariance and σ the standard deviation.

This chapter demonstrates, the all theory that proceeds of Topology Data Analysis is useful for developing algorithms based on it and get promising results in the task of emotion classification.

We started defining simplicial complexes as a set of simplices that satisfies certain conditions. This concept defines an approximation to the underlying space $\mathbb{X}$ (point cloud). To complete this step, we studied the most common algebraic and geometric methods existing. The n-dimensional homology group of a topological space and the persistent homology to work with the lack of discrimination of the homology theory were introduced. Given a complex K the persistent homology measures homology by filtration during a time, so we explained some filtration techniques and the algorithm applied in this thesis to calculate it. To summarise the information, we faced the

persistence diagrams techniques and their approaches to study from a probabilistic and statistical point. To get a description of the persistent homology as a number, the entropy persistent technique is tackled in this chapter.

A time-varying system was defined and how to combine the techniques from TDA as persistent homology and time series. To calculate the distance between time-series under certain restrictions, we introduced the Dynamic time warping technique. For extracting topological information from time-series, we studied Takens' embedding theorem and its conditions. By last, this chapter developed a background in Machine learning techniques and the different techniques that later we will apply in the experimentation chapter.

Once the background needed to understand the rest of the thesis is given, let us explain, in the next chapter, the flow work composed by different algorithms that this thesis follows.

4 Topological audio-visual emotion recognition

In this chapter, we develop an efficient method for emotion recognition, combining facial landmark points from video signal and voice from audio signal, using persistent homology tools and machine learning techniques. In order to get an efficient combination, we set a topological model to get competitive accuracy in classification task using video from emotions. Later, another topological model is built to face the classification task in audio signals of emotions. Finally, we combine both audio and video signals and propose a method that follows the strategies One-vs-Rest and One-vs-One for multi-class classification.

4.1 A topological model for video signals

In this section, we introduce the construction of the cell complex K which represents the input video (i.e., the image sequence) with a person expressing emotions. For that, we will concatenate the topological information computed from the video signals mentioned in the section above.

We start the procedure by extracting the landmark points on the face in each frame of the input video sequence as it is shown in Figure 4.1. With the intention of a fair comparison with other state-of-the-art methods, we will use the videos provided in the RAVDESS dataset [74].

Secondly, using the spatial positions of the landmark points in one frame, a cell complex, with vertices being the landmark points, is computed. In this thesis, this cell complex is computed from a Delaunay triangulation with vertices being the landmark points.

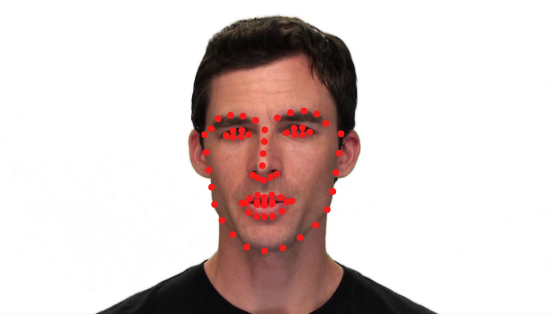

Figure 4.1: Example of computation of the landmark points of a face in one frame of an image sequence obtained from a video of the RAVDESS dataset.

Thirdly, landmark points with the same label in consecutive frames are joined by an edge. A 2-dimensional cell is obtained when the two endpoints of an edge are joined to the two endpoints of the corresponding edge in the neighbor frame. A 3-dimensional cell is obtained when the vertices of a triangle are joined with the vertices of the corresponding triangle in the neighbor frame. See Figure 4.2, where the cells of the 3-dimensional cell complex K are pictured all in the same frame.

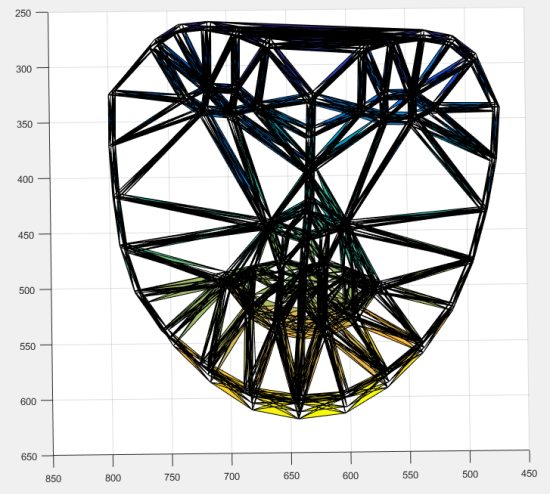

Figure 4.2: Cells of the 3-dimensional cell complex K obtained from an input video with set of vertices drawn in the same frame.

4.1.1 Filtration of the cell complex

The next step in this process is to sort the cells of K in order to obtain a *filtration*. The partial ordering of these cells is dictated by a function $f : K \to \mathbb{R}$, satisfying that if a cell σ is a face of another cell σ' in K then $f(\sigma) \leq f(\sigma')$ (i.e., σ appears before or at the same time that σ') in the ordering.

In this work, eight different filtrations (two horizontal, two vertical, and four obliques) are used to obtain eight different persistence barcodes similar to the method explained in [68, 69]. This way, all the small movements of the landmark points through the video sequence will be captured in the persistence barcodes. For each plane π, it defines the filter function $f_\pi : K \to \mathbb{R}$ which assigns to each vertex of K its distance to the plane π, and to any other cell of K, the biggest distance of its vertices to π. Ordering its cells according to the values of the function, the filtration K_π for K associated with the plane π is computed.

Lamar et. al. [68, 69] presented a version of the same procedure for gait recognition where a bunch of simplices is added to the filtration, all the simplices of $\partial K(I)$ with equal distance to the reference plane. The method is robust to variation in the number of simplices, thanks to this way in which the different times define sets of simplices with different cardinalities, and it is robust to noise.

4.1.2 Persistent homology and topological signature

The topological signature of the landmark- point face videos is obtained by computing the persistent entropy of each filtration. Let us notice that if $p_j \leq 1$, then $log(p_j) \leq 0$ and the entropy of a persistence barcode is always positive. Intuitively, the entropy measures how different intervals of the barcodes are in length.

The algorithm to compute the persistent homology is described in Algorithm 1.

Intuitive idea of Algorithm 1: The output of Algorithm 1 is the persistence diagram and, for that, the filtration of the complex is represented in the following way: A vertex, where two or more line segments meet is defined by a numerical value. The edges and triangles are defined by the index position of the face which forms it in the filtration used.

Let us describe an example of the complex formed by a triangle $1, 2, 3$ denoted by 123 and let us consider the filtration defined as: $\{1, 2, 3, 12, 13, 23, 123\}$. The index 3

Algorithm 1: Computing persistent homology (Algorithm 2 of [45]).

Input: An ordering of the cells of a cell complex K respecting a given filtration $\emptyset = K_0 \subset K_1 \subset K_2 \subset \cdots \subset K_n = K$ and boundary relations being index(σ_i) the largest index in the filtration satisfying that $\sigma_i \in K_{\text{index}(\sigma_i)}$.

Output: The persistence barcode B.

Initialize $H \leftarrow \emptyset$ and $f(\sigma_i) \leftarrow 0$, for $1 \leq i \leq m$, and $B \leftarrow \emptyset$.
for $i = 1$ **to** m **do**
  if $f\partial(\sigma_i) = 0$ **then**
    $H \leftarrow H \cup \{\sigma_i\}$ (a new homology class was born) $f(\sigma_i) \leftarrow \sigma_i$;
    $B \leftarrow B \cup \{(\text{index}(\sigma_i), \infty)\}$;
  if $f\partial(\sigma_i) \neq 0$ **then**
    Let $\sigma_j \in f\partial(\sigma_i)$ such that $j = \max\{\text{index}(\mu) : \mu \in f\partial(\sigma_i)\}$;
    $H \leftarrow H \setminus \{\sigma_j\}$ (an old homology class died);
    foreach $x \in K$ *such that* $\sigma_j \in f(x)$ **do**
      $f(x) \leftarrow f(x) + f\partial(\sigma_i)$.
    $B \leftarrow B \setminus \{(\text{index}(\sigma_j), \infty)\} \cup \{(\text{index}(\sigma_j), \text{index}(\sigma_i))\}$

is the edge with a numerical value of 12 represented by one vertex of numerical value 1 in the index position 0 and another vertex of numerical value 2 in the index position 1. See Table 4.1. To obtain this representation used the function, complex2matrix.m[1] that we described above, which has as input one of the 8 filtrations used on this thesis.

Finally, the topological signatures are used to feed a matching learning process (such as a support vector machine) to predict emotions. The full methodology is illustrated in Figure 4.3.

4.2 A topological model for audio signals

Topological data analysis has been shown to be a powerful tool for analyzing a complex data set. Tools such as persistent homology have accomplished a new method to explore the topological features and the shape of data. This method is motivated

[1]The code developed can be found in `https://github.com/Cimagroup/TFM_AudioVisual-EmotionRecognitionthroughTDA`

Index	Filtration			
0	1	1		
1	2	2		
2	3	3		
3	12	0	1	
4	13	0	2	
5	23	1	2	
6	123	3	4	5

Table 4.1: An example of how the faces of a simplex are indexed in the filtration.

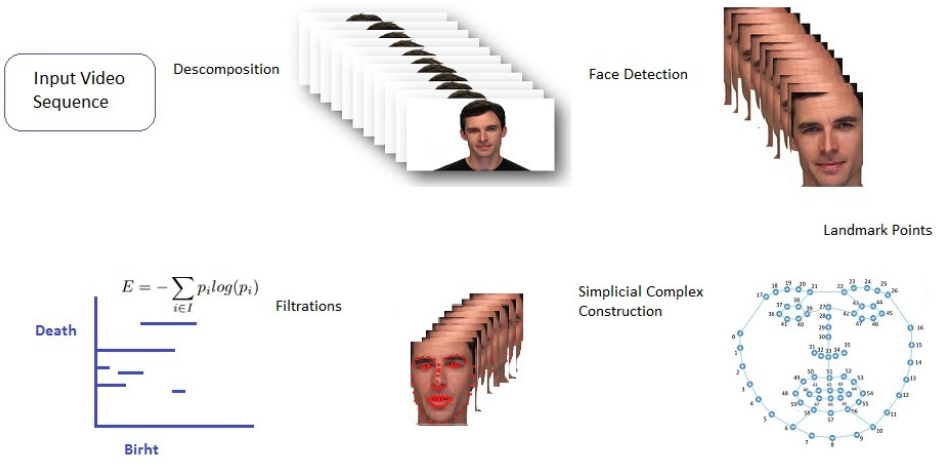

Figure 4.3: Scheme of the methodology followed.

by the development of efficient computational techniques, where consider larger and more realistic data-sets is possible. For example, methods which develop linear-size approximations of Vietoris-Rips filtration and efficient construction of the persistence diagrams [116]. A recent tool in this area called persistent entropy has been successfully applied to distinguish discrete piecewise-linear functions [105]. There is a lot of researches focus on exploring how topological information can be used to train representations enriched with geometry and topology for machine learning [94]. Thanks to these advances, TDA has been converted into a very challenging setting in the context of time-series analysis. Our goal is to explore topological features with respect to optimal delay-embedding, approximations, and time-series learning tasks.

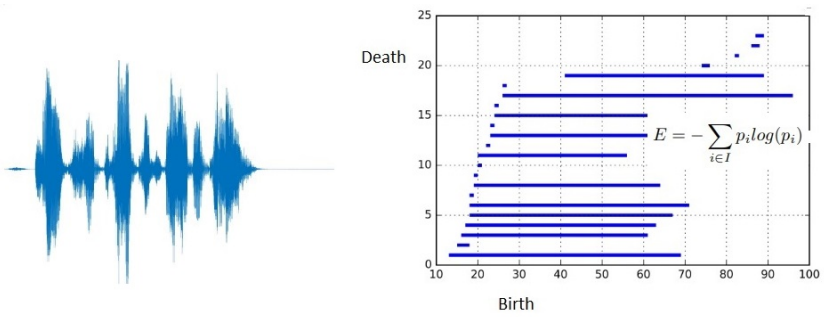

Figure 4.4: Left: An example of the audio signals considered for emotion recognition. Right: The persistence barcode obtained from the lower-star filtration of the simplicial complex computed from the audio signal by which persistent entropy is computed.

Recently, in [44], the authors developed a first topological approach to emotion recognition using just audio signals to model arousal (i.e., emotional state), considering the audio signals of different actors recorded pretending different emotions. Such approach is provided based on computing persistent entropy of the persistence homology obtained from the lower-star filtration of the signals. See Figure 4.4. Then, persistent entropy is computed to obtain a single value for each signal. Within these values, the authors applied a support vector machine to the emotion classification problem and predict emotions. According to the literature studied, no topology approaches had been previously applied to emotion recognition. This previous work [44], constitutes our starting point to face the audio emotional recognition task from a topological point of view in order to increase its accuracy. Let us now explain in detail how such methodology works:

- Subsample the signal: The size of each signal is reduced in order to face the complexity of the persistent homology algorithm.
- Embed time series into a point cloud and construct sliding on a window.
- Use Vietoris-Rips filtration on the window to have a structure encoding the geometrical shape of each window.
- Extract the relevant features of this window using persistent homology.
- Use persistence diagram to gather the information extracted
- Apply persistent entropy to summary the information.
- Support vector machine classification: This step consists of the application of several support vector machines with different kernels in order to obtain results and to develop a classification predictor for emotions. The different possible kernels are tested and the one with better accuracy is chosen.

In the case of audio signals, in this thesis, we will focus on defining how to spread out the points relative to each others concerning a "distance" previously defined. This set focuses on the connectivity between points based on the numerical value at each point. Similar to the analysis about 0-persistent homology, where increasing a threshold, we track the connected components, now the analysis focuses on the horizontal distribution between points. Intuitively, it is like scanning a line up over the signal. See Figure 4.5.

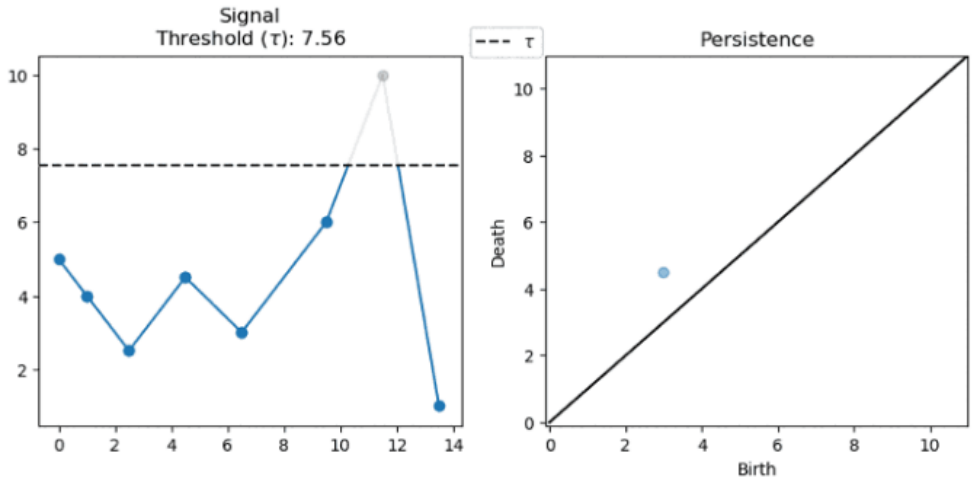

Figure 4.5: Left: example of a signal. Right: the corresponding persistence diagram.

We can think of persistent homology as a pattern of a static object. Persistent homology has been applied to time-varying systems considering continuous representations [23], or introducing statistics evaluating in time the system developed [122]. In this work, we combine persistent homology and the notion of time series, to characterize the evolution of a variable geometry space in time.

A time series is a collection of data obtained from different observations in time. According to their generality and flexibility, time series are very useful, applications such as classification and segmentation are supported by a theoretical framework. The idea is to capture changes of a variable-geometry space as a time series of persistence diagrams, and later compare these time series by using dynamic time warping.

4.2.1 Time series as point clouds

The usual setting in TDA is to generate a simplicial complex from a point cloud. Now, we are facing a time series. Let us see how we can obtain that point clouds. We assume that the audio signal is a discrete-time series, visualized as scatter plots in two dimensions. It is easy to analyze the local behavior of time series with the technique developed in this field like discrete Fourier transform to know if the signal in a window arises as to the sum of few simple periodical signals.

In this work, a different way to encode a time-evolving process is presented in order to analyze the effects of its dynamics, based on the idea that these occur in higher dimensions. The idea is to represent the time series as a set of vectors in a Euclidean space of arbitrary dimension. To proceed, we need to choose two integers m and τ. Then, for each time $t_i \in (t_0, t_1, \dots)$, the values of the variable y at m different times are collected, evenly spaced by τ and starting with t_i, forming a vector with m entries:

$$Y_{t_i} = (y_{t_i}, y_{t_i+\tau}, \dots, y_{t_i+(m-1)\tau}).$$

The result is a set of vectors in an m-dimensional space; as we described in the previous sections, the technique used is Takens' embedding, where, m is the embedding dimension and τ the time delay parameter. Finally, using this procedure, we obtain a time series of points clouds with interesting topology features inside, ready to study. The optimal selection of these parameters for the dataset used will be described in Chapter 5.

4.2.2 From point clouds to persistence diagrams

After generating time series from point clouds, it is needed to obtain information from them. Persistent homology is applied in order to extract the topological features that persist over time, providing a concise description of the topological changes overall scales of the data. This information is obtained through a filtration previously defined on the data that captures the birth and death of the topological features across dimensions such as connected components, tunnels, voids. We used the Vietoris-Rips filtration to generate a linear size filtration, which is used in the computation of the persistence diagrams [116]. Expressly, we focus on exploring the 1-dimensional persistence diagrams, which provides the information of the 1-dimensional topological features found in the data, interpreted as cycles, in order to characterize periodicity and repetitive patterns located in time series data.

Then, given two windows and their corresponding persistence diagrams, it is possible to calculate a variety of distance metrics in order to compare them with respect to topological similarity. The distance metrics for persistence diagrams affords relate the topology of the dataset in topological terms.

4.3 An audio-visual combination topological model

Once the topological models for each kind of information have been defined, it is overriding to develop a methodology to combine them. As we see in the previous sections, both models are focused on the obtention of persistent entropy value for each filtration defined. These values will be used to build a vector of features. The first eight features are associated with the filtrations defined in each direction from the video signals, and the last one is obtained from the audio topological model.

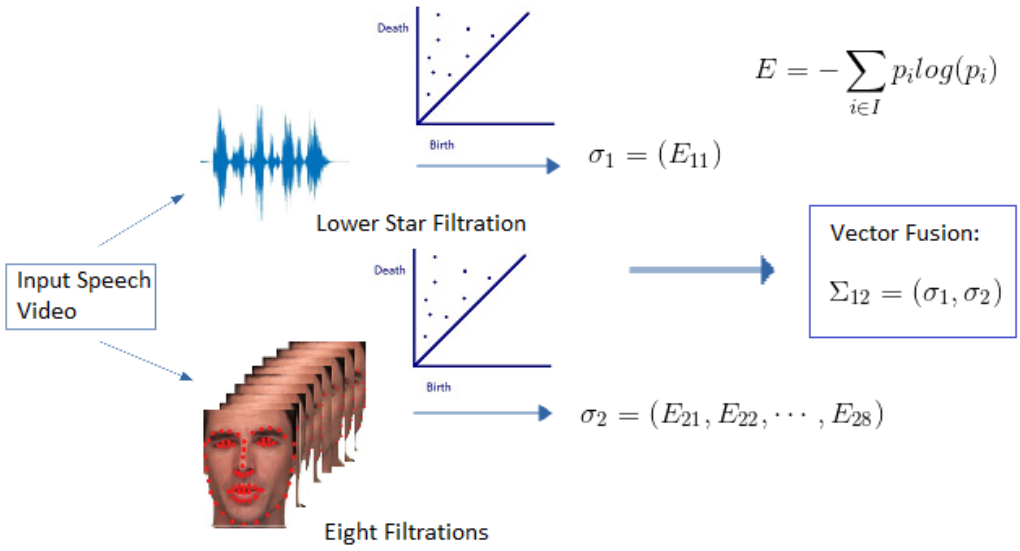

Figure 4.6: Construction of a 9-dimensional fusion feature vector. Position 1 of the vector is associated with the persistent entropy obtained from the lower-star filtration [105] of the raw audio signal. The rest of the positions are associated with the eight filtrations obtained from the eight fixed planes.

Specifically, the eight persistent entropy values associated with the eight filtrations together with the persistent entropy previously computed for the audio signals

from the topological signature of the input video which consists of a vector in the 9-dimensional space $\mathbb{R}^9$ (see Figure 4.6).

In the literature, there are different approaches to tackle the multi-classification problems. To split the multi-class classification dataset into multiple binary classification datasets and set a binary classification model on each we will follow the next two strategies: "One-vs-Rest" and "One-vs-One". Consult the next chapter to deepen in knowledge about these methods.

In this chapter, we have explained the algorithms developed in this thesis with a theoretical focus detailing some parameters that in the implementation are necessary to research. The next chapter focuses on an aspect more practical of our workflow, with the results of the experimentation work.

5 Experimentation

In this chapter, the experiments that support the effectiveness of the use of topological signatures on video and audio signals for the emotion recognition task are exposed. Hereafter, Section 5.1 addresses the database used in the experiments. The protocol for the experiments on the video-only dataset is explained in Section 5.2, and Section 5.3 focuses on the protocol for the experiments on the audio dataset. Finally, we explain the configuration followed for combining both datasets in Section 5.4.

5.1 Datasets description

For experimentation, the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) is used [74]. This database contains the vocalization of two statements in a neutral North American accent by 24 professional actors (12 female, 12 male). Each expression is produced at two levels of emotional intensity (normal, strong), with an additional neutral expression. The intensity fulfils an important role in emotional theory [28, 107]. The strong intensity is useful when we are looking for clear emotional examples. However, the normal intensity is used if we are interested in providing classification for daily life [59].

The RAVDESS database [74] is based on multi-modals inputs: physiological sensors, video, depth sensors, and standard input devices, providing more information to the recognition process. Because of the information it provides ,it is widely used in emotion recognition.

The RAVDESS database is composed of 7356 files. All actors produced 104 different vocalizations, consisting of 60 spoken expressions and 44 sung expressions. These vocalizations are available in three formats: audio-video, video-only, and audio-only. This produced 312 files per actor forming 7356 files with a total of 4320 speech record-

ings and 3036 song recordings. For this thesis, we are going to work with the video-only dataset and the audio-only dataset. Some examples of the RAVDESS database can be seen in Figure 5.1.

5.1.1 Video-only dataset

The RAVDESS video dataset contains tracked facial landmark movements for all 2452 trials. The information offered includes facial landmark detection, head pose estimation, facial action unit recognition, and eye-gaze estimation. All the information is collected on 2452 CSV files. Each actor has 104 tracked trials (60 speech, 44 song).

File naming convention

Each of the RAVDESS files has a unique filename describing different features. The filename consists of a 7-part numerical identifier. In the next experiments, we are going to focus on the modality "video-only" represented by 02 in the first digits of the identifier. Then, in order, the identifiers are the following:

- Modality (02 = video only)
- Vocal channel (01 = speech)
- Emotion (01 = neutral, 02 = calm, 03 = happy, 04 = sad, 05 = angry, 06 = fearful, 07 = disgust, 08 = surprised.)
- Statement (01 = "Kids are talking by the door", 02 = "Dogs are sitting by the door").
- Repetition (01 = 1st repetition, 02 = 2nd repetition).
- Actor (01 to 24) Odd numbered actors are male, even numbered actors are female.

Filename example: `02-01-04-01-02-13.CSV`
Meaning the following:

`02`	Video-only
`01`	Speech
`04`	Sad
`01`	Statement "Kids are talking by the door."
`02`	Second Repetition
`02`	14th Actor Male, as the ID number is odd

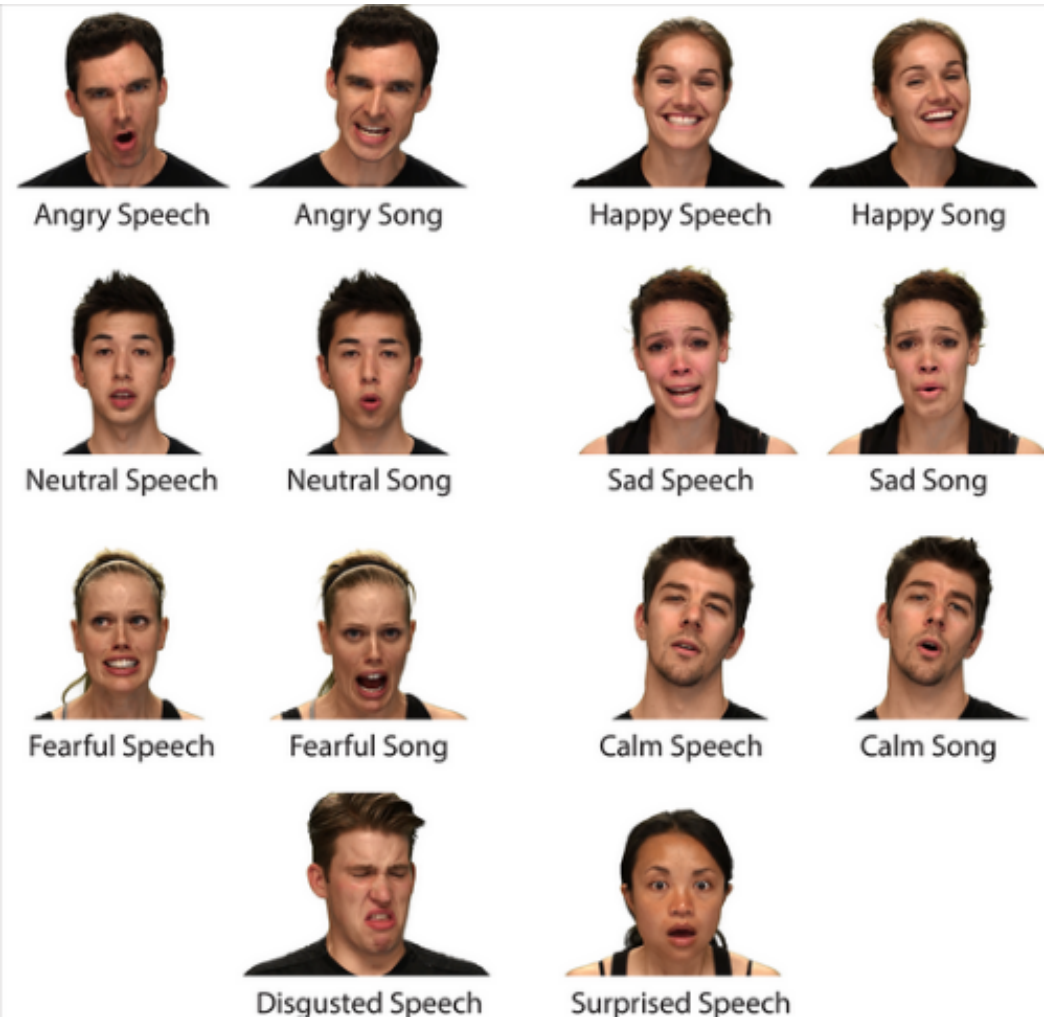

Figure 5.1: Example of the eight RAVDESS emotions of audio dataset (speech channel).

5.2 Experimentation on video-only dataset

The first experimentation consists of the computation of a simplicial complex from the facial landmarks of nine frames of each video of the RAVDESS dataset. Here, we hypothesize that summarising an emotion video by a set of a few key-frames in terms of the variability of facial expressions will be enough to describe and efficiently learn the contained emotion. Then, we proceed with the application of Algorithm 1 in page 50 using eight different filter functions from which we can compute their persistent entropy values. More details on the implementation can be consulted in Section 6.

We focused on the 60 speech videos collected in the RAVDESS dataset. The total tracked files used is 24 actors × 60 speech trials, i.e., a total of 1440 files. In the RAVDESS dataset, the tracking results are provided as individual CSV files with values separated by comma. The resolution of all input videos is 1280×720, the output units are in pixels and their range of values goes from $(0, 0)$ (top left corner) to $(1280, 720)$ (bottom right corner). Figure 5.2 shows the landmark points on a face.

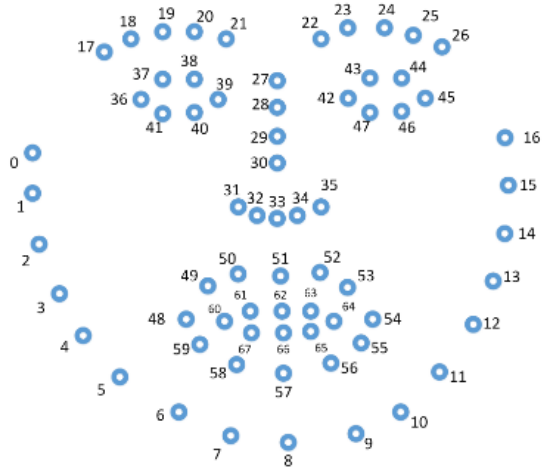

Figure 5.2: Location of 2D landmark points.

5.2.1 Classifying emotions using the facial landmarks

This subsection focuses on the behavior of the 62 landmarks points of the face around the video from one emotion. The first task in the experimentation topic is which classifier to select.

In the literature, several pattern classifiers are explored for developing speech systems like speech recognition, speaker recognition, emotion classification, speaker verification, and so on [27, 112]. Many researchers explored several classification methods, such as neural networks [133], gaussian mixture model [13], kernel regression [88], k-nearest neighbors [24] and support vector machines [73]. Each classifier has advantages and limitations over the others. In this thesis, after making some tests with different classifiers and according to the characteristic of our database, we decided to apply the k-nearest neighbor family of classifiers which typically have good predictive accuracy in low dimensions. Such classification uses feature similarity to predict the feature of new data points.

A feature will be assigned to a new data point based on how closely it is from the points in the training dataset. It is possible to use different metrics to determine that distance. Given a set of points $X \in \mathbb{R}^n$ and a distance function, the k-nearest neighbor (kNN) method finds the k closest points in our dataset to a query point or set of points. kNN-based algorithms are widely used as benchmark machine learning rules 3.6.1. The classifiers with better output for our data were **Fine KNN**, which distinguishes finely in detail between classes. The number of neighbors was set to $k = 1$, which means that the feature is simply assigned to the class on the single nearest neighbor.

Evaluation Procedure

1. We need two kinds of attributes in the data: the columns associated with the features and one column with the label.
2. To understand the model performance, we divide the dataset into a **training set** and **test set**.
3. Then, we build a KNN classifier model for $K = 1$ using the library and functions explained on 6.
4. We estimate how accurately the classifier can predict the type of emotion.
5. Accuracy is computed by comparing actual test set values and predicted values.

The dataset was randomly split in 20% of the observations for the test set, and 80% for the training set. In this experiment, each point of the dataset consists of a vector of eight features obtained from the eight filtrations applied to each video. These features are the persistent entropy values previously calculated in the algorithm.

Table 5.1: Accuracy rates reached using the entropy values associated with one filtration

Acurracy	Filt1	Filt2	Filt3	Filt4	Filt5	Filt6	Filt7	Filt8
Training_Data	88,7%	81,3%	84,93%	89,7%	85,6%	87,3%	82,7%	85,4%
Test_Data	86,2%	79,0%	83,5%	88,3%	84%	86,2%	81,1%	83,4%
Total_Data	89,6%	80,7%	87,0%	89,9%	88,1%	86,1%	84,4%	87,0%

The **first experiment** focuses on classify the data using only the persistent entropy values associated with one filtration. Since we defined eight different filtrations, we will have eight classifications associated with them. Table 5.1 offers the accuracy

rates reached in both parts of the dataset, and the last row shows the accuracy of the trained model on the 100% of the dataset.

The first direction Filt1, horizontal (axis X) going from bottom to top, and the fourth one Filt4, going orthogonal to the third direction of view (45 degrees with X and Y) reach better accuracy than others. However, the classification ranges in each filtration are quite similar with a mean of $86,6\%$ in the analysis of the full dataset. Once this experiment is done, it allows to confirm that every filtration offers relevant information for the emotion classification. Then, a vector of eight dimension is built that groups each persistent entropy value associated to each filtration defined.

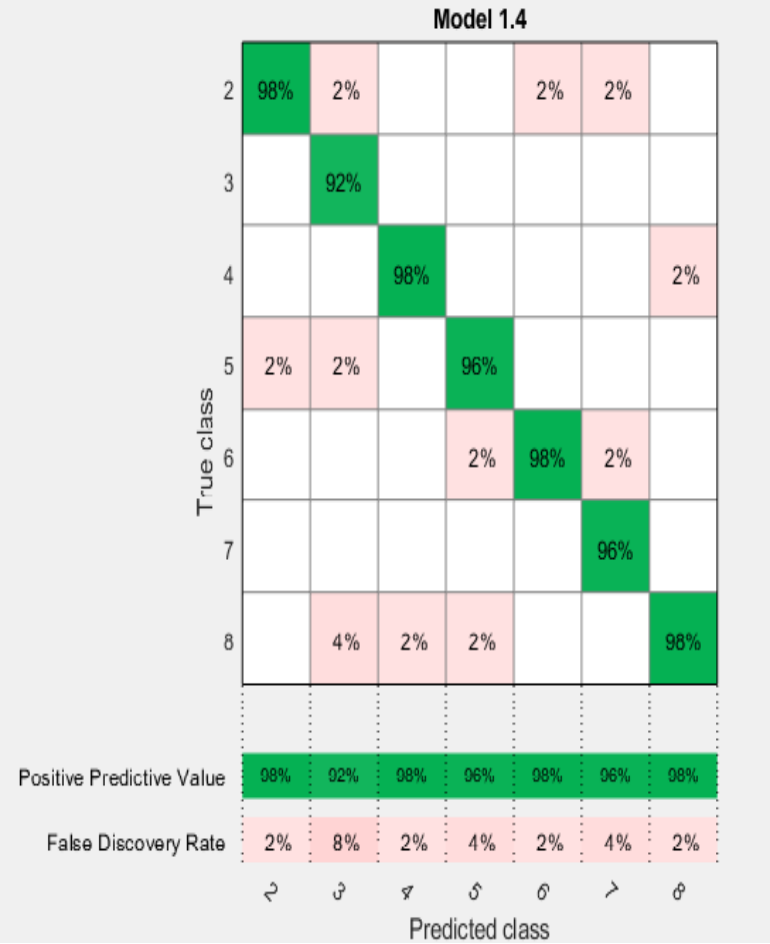

Figure 5.3: Confusion Matrix.

The **second experiment** on this dataset focuses on mixing the information of the eight filtrations. Then, the dataset comprises vectors of eight features with the same

target (the type of emotion) as the first experiment. Figure 5.3 shows the confusion matrix plot associated to the training set.

This matrix allows us to understand how the kNN Fine classifier performed in each class, identifying the areas where the classifier has performed poorly. The numbers mean emotions: 2 = calm, 3 = happy, 4 = sad, 5 = angry, 6 = fearful, 7 = disgust, 8 = surprised. As we can see, the happy emotion has a lower rate than the others. Then, movements and gestures that people make when expressing happiness in the video-dataset analyzed are less significant to characterize such emotion.

Table 5.2: Accuracy rates reached on video signals.

Emotions	Training Data	Test Data	Full Data
calm	98%	94%	99%
happy	92%	90%	95%
sad	98%	95%	99%
angry	96%	93%	98%
fearful	98%	96%	99%
disgust	96%	94%	97%
surprised	98%	97%	98%

Table 5.3: General Accuracy rates reached on video signals.

Total	Training Dataset	Test Dataset	Full Dataset
General Accuracy	**96,57%**	**94,14%**	**97,85%**

On the training dataset, the accuracy was reached 96, 4%, reflected on the matrix. On test dataset, 94, 1% was reached. And, on the full dataset, 97, 8% was achieved. These rates of accuracy allow us to claim that the topological information extracted from the action in the video signals is relevant and decisive to classify emotions. Table 5.2 and Table 5.3 shows a summary of this experiment with the accuracy achieved in the video-only dataset.

5.3 Experimentation on audio dataset

The second experimentation is focused on analyzing raw signals of the audio dataset from RAVDESS. In this case of audio signals, the dataset is composed of 24 actors

interpreting 60 audios of different emotions. Each person contains four audios representing neutral emotions and eight audios for each of the seven remaining emotions.

5.3.1 Experiment 1: audio dataset

Figure 5.4 shows examples of the raw signal from each emotion. We can see the differences in intensity, amplitude, and information which contain each audio signal.

For the **first experiment**, we follow the next idea: For each audio, we partition the raw signal into a series of segments and compute automatically the optimal delay coordinate embedding. Each stage is totally configurable, where the segment size, window size, delay parameters should be controlled. The choice of the embedding dimension m and time delay τ determine the number of points in the point cloud and it is an important step in the algorithm (see subsection 5.3.1). For more information, Section 6 includes the explanation of the codes for the computation of these parameters.

How to choose the delay and the embedding dimension optimal for this dataset? We do not know these parameters a priory. Consider two or three embedding plots should be the more realistic case, but we need to formalize this. Ín the literature, we found two measures that give an idea about which delay and dimension to choose. Next, we will see the explanation and reference of these measures.

Choice of τ and m

Mutual Information [127] is a measure of similarity between two labels of the same data, where we need to calculate the minimum x_{min} and maximum x_{max} of the time-series analyzed. Later, the interval $[x_{min}, x_{max}]$ is divided into a greater number of pieces. U_k denotes the probability of one element of the time series is in the k-th piece and $U_{h,k}(\tau)$ the probability that x_i is in the k-th piece when $x_{i+\tau}$ is in the k-th piece. Then the mutual information is formulated by:

$$I(\tau) = -\sum_{h=1}^{N}\sum_{k=1}^{N} U_{h,k}(\tau) log \frac{U_{h,k}(\tau)}{U_h U_k}.$$

The first minimum of $I(\tau)$ as a function of τ gives the optimal delay, since there we obtain the largest information by adding $x_{i+\tau}$.

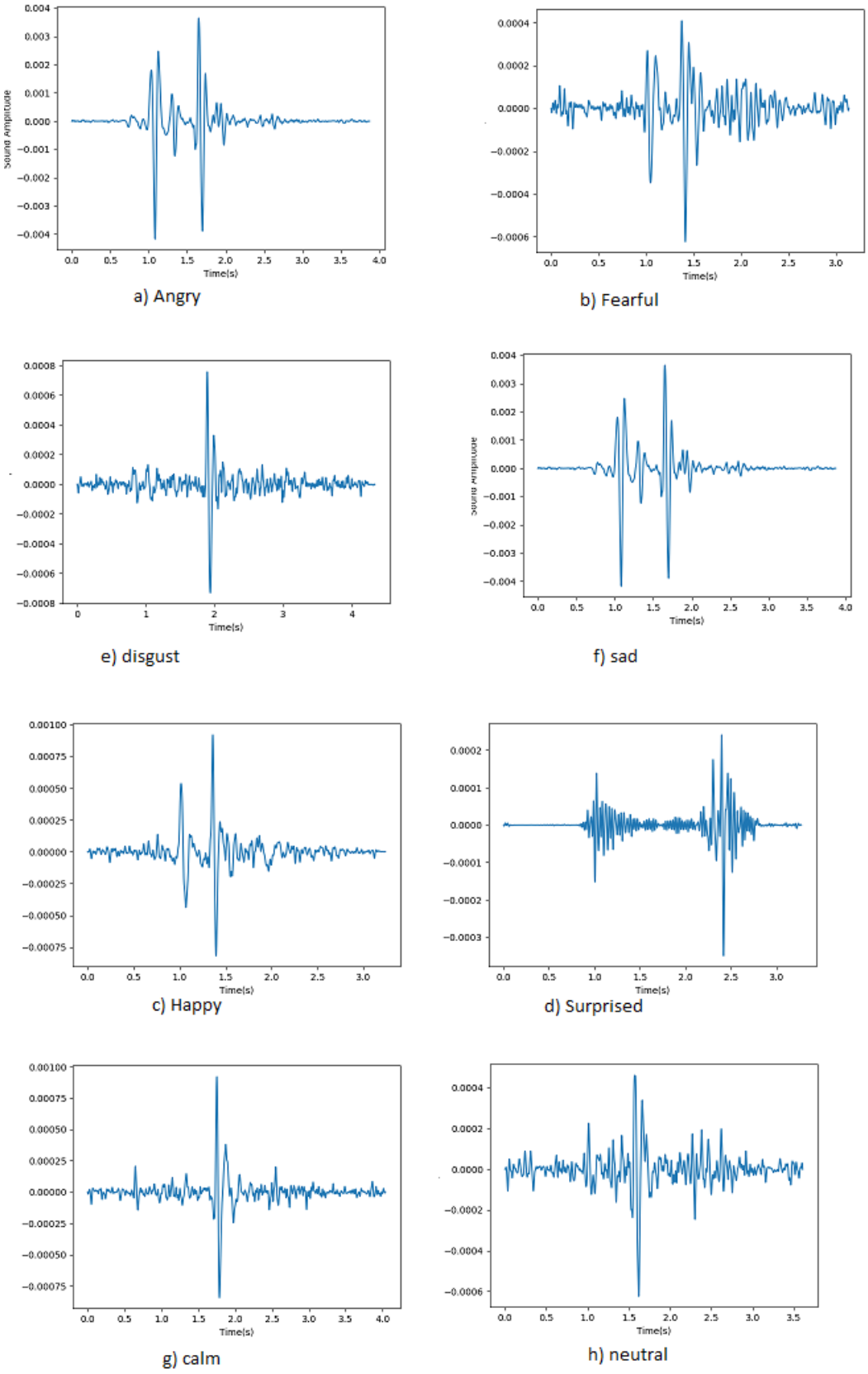

Figure 5.4: Examples of the raw signal from each emotion.

Then, we calculate the mutual information for each audio data. The number of pieces chosen is 10 since we have 1000 data points obtained after applying a technique of re-sampling, so we can expect to have around 100 points in each piece, and the audios are approximated of 5 seconds.

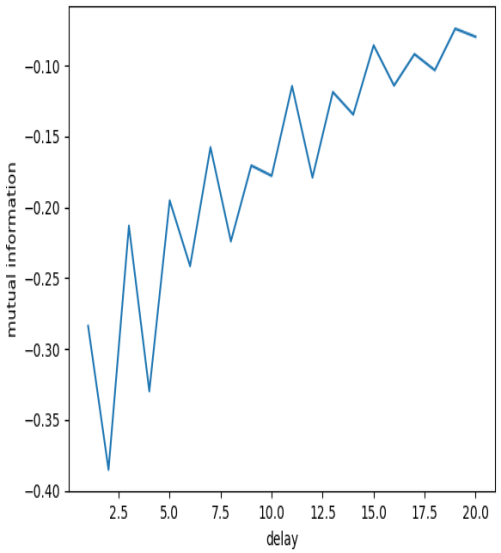

Figure 5.5: Mutual Information.

As we see in Figure 5.5, the mutual information is calculated for the dataset given τ, and we get its first minimum. This gives us the optimal embedding delay; since the plot of the mutual information has an increasing behave, the optimal delay selected is 1. This plot should look different if the data sample has a higher frequency or with more data points. A sufficiently large time delay window is an important issue for a time series predictor. If the window is too small, the attractor of the system would be projected into a space of insufficient dimension, in which the proximity is not related to the actual proximity on the original attractor. Then, for that, it is important to control this parameter [14].

To determine the correct embedding dimension, we are going to use the measure known as *false nearest neighbors*, the most popular tool proposed by Kennel [61]. If we have a point m_i with a neighbour m_j ($||m_i - m_j|| \leq \epsilon$) for $\epsilon > 0$ then we analyzed the normalized distance D_i for the next dimension. If D_i exceeds a given heuristic threshold D_j this point is marked as having a false nearest neighbor. Then,

the embedding dimension is high enough if the points for which

$$D_i = \frac{||x_{i+\tau} - x_{j+\tau}||}{||m_i - m_j||} > D_j$$

is zero, or sufficiently small. If this is true, then we have a false nearest neighbour.

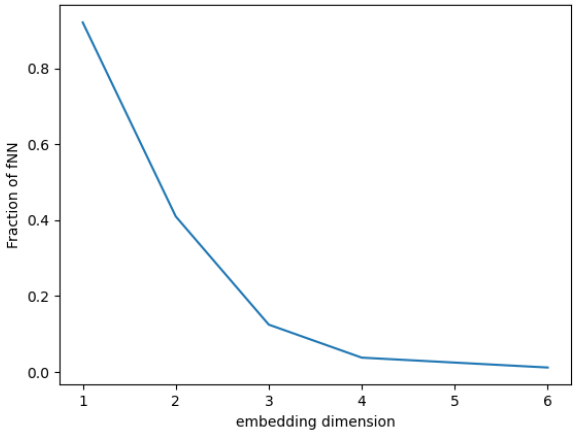

Figure 5.6: Parameters Embedding.

In Figure 5.6, we see that false nearest neighbours decline to zero when the embedding dimension is 4. Then, we decide to model the data with a system of 4 variables. It is important to remark that previous plots describe the optimal embedding parameters for one specific audio signal. Thorough research threw us that the majority of audios have the same behavior related to the selection of 4 variables for the embedding. However, the algorithm implemented calculates automatically these parameters in each iteration in the audio-signal dataset, for later applies the next steps of our model in the embedding signal obtained. The 2D and 3D plots pictured in Figure 5.7 show a visualization of the embedded data with the optimal parameter for different emotions.

For this experiment, the Vietoris-Rips filtration is applied to generate the persistence diagrams associated with the signal. These diagrams allow us to explore the behavior of the existing topological features. Finally, to explore the potential of the topological information extracted, we compute the persistent entropy of the diagrams. Then, each value of entropy obtained will be consider a point of a dataset.

We use, as the classification technique, a support vector machine (SVM) with fold cross-validation and the kernel that provides better accuracy from the ones explained

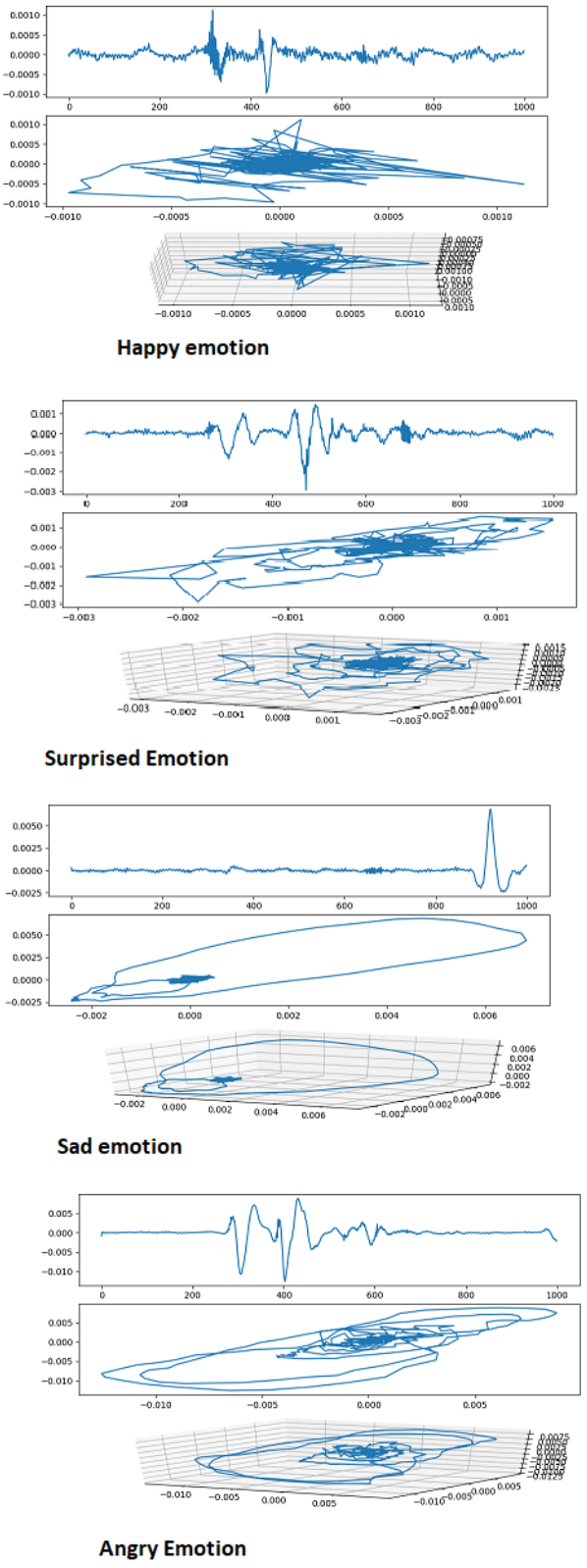

Figure 5.7: Embedded data with the optimal parameters associated (from up to down: example of one happy emotion audio signal, second surprised, third sad emotion and last one the angry emotion).

previously. In summary, an SVM classifies data by finding the best hyperplane that separates data points of one class from those of the other class. The best hyperplane for an SVM means the one with the largest margin between the two classes.

We tested in our dataset different kernel functions to specify one. The kernel that better results yielded was the Linear Kernel, making a simple linear separation between classes.

This experiment yields poor accuracy, where a $27,4\%$ is reached for all data set and on the training set $21,3\%$. Some conclusions similar to [44] can be marked from this failed experiment: the emotion recognition problem is a multidimensional one, in the sense that 1-dimensional value is not enough to obtain a good classification. Hence, from the topological point of view, emotions are reflected more clearly in facial expressions as compared to voice using.

5.3.2 Experiment 2: audio dataset

In this experiment, we make the same procedure as before, with one difference in the classification. Instead of comparing with a 1-dimensional value, "the persistent entropy", and using the SVM classifier, we applied the k-nearest neighbor classifier.

The new idea is to calculate a distance matrix among the persistence diagrams, selected the bottleneck distance for this comparison. This matrix contains the distances that have been calculated between each pair of elements possible. We divided the distance matrix in training and test sets as we did for the Video Dataset experiment. Table 5.4 displays the results for both sets. Following this technique, the accuracy increases achieving a mean of 43%.

Table 5.4: Accuracy rates reached using the distance matrix.

Emotions	Training Data	Test Data
calm	38%	26%
happy	37%	42%
sad	41%	39%
angry	43%	41%
fearful	50%	48%
disgust	46%	41%
surprised	44%	32%

Even having higher results than the previous algorithm, the accuracy keeps low. This alerts us about the incorrect idea of our method based on the theorem of Takens' Embedding. According to the theorem, the time series are expected to be driven by a non-linear system and should be guaranteed that the resulting embedding is a faithful reproduction of the original dynamics. In our case, the results display that, for our data, this assumption is not satisfied. One reason is given by these time series are not homogeneous or cyclical.

To find a method that increases the results archived until now, we present the next experiment.

5.3.3 Experiment 3: audio dataset

For this experiment, we follow the previous idea that generated a matrix distances based on the bottleneck distance among the diagrams obtained from Vietoris-Rips filtration of the signal. For this experiment, we applied some variation to achieve higher accuracy.

- **Subsampling**: The process of sampling a signal to face the complexity of the persistent homology algorithm.
- **Imperceptible noise**: Signals are slightly perturbed to fulfill the requirement of lower-start filtrations.
- **Persistence diagram**: The lower-star filtration is applied to these signals, obtaining the associated persistence diagrams.
- **Matrix distance**: Bottleneck distance between each pair of diagrams possible.
- **Classification**: This step consists of the application of k-nearest neighbors with different values of k in order to infer results and develop classification predictors to emotions. The input of the KNN-algorithm will be the matrix distance and the vector of the labels, each row of the matrix is the feature to classify.

By this, the accuracy of the method showed in Table 5.5 for classification by emotions was obtained using the KNN-nearest neighbors with $K = 5$.

After the three experiments explained above, the last one indicates better results for the task of audio emotion recognition. Considering other results in the literature and the previous work of our group [44] in audio emotion classification, we can confirm that we reached better and promising results for this task. However, we are still far from the accuracy reached on the emotional video classification.

Table 5.5: Accuracy rates reached using the KNN-nearest neighbors with $K = 5$.

Emotions	Training Data	Test Data	Full Data
calm	76,3%	74%	77%
happy	72,4%	70,2%	73,4%
sad	83%	79,6%	84,2%
angry	82,4%	81%	83%
fearful	86,5%	82,3%	88%
disgust	80,3%	77%	81,5%
surprised	82,5%	80,4%	85%

5.4 Combination of datasets

With the aim of improving the results reached in the first approach (video-only emotion recognition), one of the goals of this thesis is the idea to mix the information obtained from the two approaches (video-only and audio-only emotion recognition). The first approach was using the landmark points of the face to get the Delaunay complex and applied the methodology explained before to the eight different filtrations in order to collect all possible information from the face. This approach presents a vector of dimension 8 with the eight persistent entropy values from each video. In this experiment, we include, in the position 9 of each vector, the persistent entropy value obtained from the process described for audios. Sadly, such experiment does not offer good results, and the bad results achieved in the experiment 5.3.1, greatly affect the results achieved in this experiment combining audio with video signals.

Another approach is the following: After training classifiers following different setups by using audio and video signal, we obtain the confidence values for each emotion in every dataset. Later, we decide to work with the setups that reached better accuracy in each dataset: the second setup from the video dataset and the third one from the audio dataset. Then, the confidence value is fused to train a stacked classifier obtaining final emotion prediction with both signals. Next, we explain the methodology followed to achieve a successful combination.

Multi-class classification is not supported by some classification predictive models. Algorithms such as Logistic Regression and Support Vector Machines were created for binary classification and they not support more than two classes. To use binary classification algorithms for a multi-classification task, one setup is to divide the dataset into multiple binary classification datasets and fit a binary classification model

on each [37]. There are two famous strategies for this approach: the One-vs-Rest and One-vs-One.

One-vs-Rest: A heuristic method that involves dividing the multi-class dataset into multiple binary classification problems. Then, each binary classifier will be trained on each binary classification problem and finally, the predictions are made according to the most confident model [85].

One-vs-One: A heuristic method that involves dividing the datasets into one dataset for each class versus every other class. Then, this implies an increase in the number of datasets due we will have one for each pair of classes. If the number of classes is high, it is not recommendable to use this method in code efficiency [85].

The library *Skelarn* has implemented the One-vs-Rest and One-vs-One setups to make the multi-class classification. Both setups were tested for the combination model task and One-vs-Rest reached better results with respect to the One-vs-One approach, one example of its application will be explained in 6.

Table 5.6: Accuracy rates reached by combining audio and video signals.

Emotions	Training Data	Test Data	Full Data
calm	90,2%	86,5%	92%
happy	89,4%	87,6%	90,8%
sad	95%	93%	96%
angry	90%	88%	92,4%
fearful	93,3%	90,2%	95,3%
disgust	89%	87,5%	92,6%
surprised	88,5%	85,5%	90%

Table 5.6 displays the accuracy for the combination using One-vs-Rest setting. The approach One-vs-Rest implemented for the mixed model of information makes the predictions according to the most confident model. We can see that the previous accuracy obtained in the video-only dataset is affected by the introduction of audio information, but this decrease in values of classification is not so higher, and we reached results promising and competitive in the area. In the training dataset, the accuracy was reached $90,7\%$; on test dataset, $88,3\%$ was reached. And, on the full dataset, $92,4\%$ was achieved.

5.5 Comparison with current papers

Lately, hybrid neural networks combining Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have become the state-of-art for emotion recognition. As we argued in this thesis, audio-video emotion recognition is a challenging task. Deep learning has played an important role in most of the challenges that exist in the biometrics recognition task.

Guo et al. [49] proposed an audiovisual-based hybrid network that combines the predictions of five models for emotion recognition in the wild. The overall accuracy of the proposed method achieves 55, 61 and 51, 15 classification accuracy on the audio and video-only dataset, respectively, while the challenge baseline is of accuracy 38, 81. This paper uses a different database (AFEW database [66]) but with the same amount of emotions to analyze and with similar conditions.

Moskovin et al. [87] developed a method based on convolutional and recurrent neural networks for real-time human emotion recognition by audio-visual data. Based on a series of experiments, the optimal hyper-parameters of the neural network, as well as parameters of data processing, were chosen. The results achieved for the emotion classification in working with the only audio channel is of 61%, with video channel 69%. To fuse the results of two neural networks is 73%, they showed the information only in the full dataset. They use the Acted Facial Expressions from the Wild Database [79].

Issa et al. [55] face only the audio emotion recognition task using a convolutional neural network. Their baseline model includes one-dimensional convolutional layers combined with dropout, batch normalization, and activation layer.

Table 5.7 shows how our method behaves with respects to principal papers from state-of-art. The values showed are achieved on the training dataset to present the results as the papers compared.

Table 5.7: Comparison of our method to start-of-the-art methods

Paper	Audio dataset	Video-only dataset	Audio-video dataset
Guo et al. \[49]	55,61%	51,15%	38,81%
Moskovin et al. \[87]	61%	69%	73%
Issa et al. (Model E) [55]	**86,1**%		
Our method	80,48%	**96,6%**	**90,7%**

We can conclude that our method increases the previous better accuracy reached in this area, except in the case of the audio signal, that we need to work on to increase our results.

More information about the functions, library, software, and algorithms used in this thesis is explained in the next section.

6 | Design and Implementation

This chapter will explain the use of different programming languages and certain libraries applied in this thesis, as well as discuss the implementation of the complete system. The code can be consulted in https://github.com/Cimagroup/AudioVisual-EmotionRecognitionUsingTDA.

6.1 Analysis of the implementation of the algoritm for the video-only dataset

According to the first issue of the thesis, the useful functions existing, and the previous works focused on this issue, we have chosen the language Matlab to code the first part.

The speech videos collected in the RAVDESS dataset are provided as individual CSV files with coordinates of the facial landmark points separated by a comma.

The first issue of the algorithm was to define a point cloud $\{(x_i, y_i)\}$ and apply the 2D Delaunay Triangulation, a function offered by Matlab. Each element in *DT.ConnectivityList* is a vertex ID and each rows represents a triangle in the triangulation. Then, we applied the function *triangulation* to create the 2D triangulation data in matrix format.

```
1000 DT = delaunayTriangulation(x,y)
     %Connectivity list
1002 list = DT.ConnectivityList
     # creates a 2D or 3D triangulation representation using the
        triangulation connectivity list DT and the points in matrix P.
1004 Delaunay_base =  triangulation(list, DT.points)
```

Listing 6.1: Delaunay Triangulation.

Now that our data is represented using triangulation, we can perform topological and geometric queries. The algorithm uses this representation to built the topological complex. It takes the landmark points with the same label in consecutive frames and then joined by an edge (worked with 9 frames).

The next function defined in the thesis calculates the eight filtration of the cell complexes obtained from the four view directions (2 by each direction.)

```
function filters = filtrations(complex)

    pts_central=[];
    n=size(complex.pts,1);
    %n= max(complex.cuad(:,4))
    for i=1:len(complex)
      pts_central(i,:) = sum(complex.pts(complex.cuad(i,:),:))/4;
    end

    maxP = max(max(abs(complex.pts))) + 10000;

   %sort triangles

    [X,indX] = sortrows(pts_central); %ordering the points by the
   elements of the first column

    listPtsY(:,1) = pts_central(:,2); %reverse the order of x and y
    listPtsY(:,2) = pts_central(:,1);


    [X,indX] = sortrows(pts_central);
    [Y,indY] = sortrows(listPtsY);

     PtsDistXY = arrayfun(@(x,y) (abs(x+y)), pts_central(:,1),
   pts_central(:,2));

     tem = PtsDistXY;
     tem(:,2:3) = pts_central;
     [XY,indXY] = sortrows(tem);

     PtsDistYX = arrayfun(@(x,y)  (abs(x-y + maxP)/sqrt(2)),
   pts_central(:,1),pts_central(:,2));

     tem = PtsDistYX;
     tem(:,2:3) = pts_central;
     [YX,indYX] = sortrows(tem);


```

```
      complexX = complex.cuad(indX,:);
1036    subcomplex =  complexX; %to invert order
        ind = [size(subcomplex,1):-1:1];
1038    subcomplex = subcomplex(ind,:);
       complexX_inv = subcomplex;
1040
       complexY = complex.cuad(indY,:);
1042    subcomplex =  complexY;
        ind = [size(subcomplex,1):-1:1];
1044    subcomplex = subcomplex(ind,:);
       complexY_inv = subcomplex;
1046
       complexXY = complex.cuad(indXY,:);
1048    subcomplex =  complexXY;
        ind = [size(subcomplex,1):-1:1];
1050    subcomplex = subcomplex(ind,:);
       complexXY_inv = subcomplex;
1052
       complexYX = complex.cuad(indYX,:);
1054    subcomplex =  complexYX; %reverse order
        ind = [size(subcomplex,1):-1:1];
1056    subcomplex = subcomplex(ind,:);
      complexYX_inv = subcomplex;
1058
      filters.complexX = unique(complexX,'rows','stable');
1060  filters.complexY = unique(complexY,'rows','stable');
      filters.complexXY = unique(complexXY,'rows','stable');
1062  filters.complexYX = unique(complexYX,'rows','stable');

1064  filters.complexX_inv = unique(complexX_inv,'rows','stable');
      filters.complexY_inv = unique(complexY_inv,'rows','stable');
1066  filters.complexXY_inv = unique(complexXY_inv,'rows','stable');
      filters.complexYX_inv = unique(complexYX_inv,'rows','stable');
1068 end
```

The next function is defined with the aim to split the cell complex obtained to see the features which born and died 2-cell on the complex. Later, the components which survived get the index where the square is.

```
1000
  function[index_cuad, complejo] = complex_wtsquare(complejo)
1002
  index_cuad=[];
1004 for i=1:length(complejo)
```

```
    if numel(complejo{i})==4
1006    index_cuad = [index_cuad i];
    end
1008 end
  complejo(index_cuad) = [];
```

After we get the eight filtrations in each direction defined, we applied the next function to convert the complex in a matrix way. Once we had the matrix, we apply the Incremental Algorithm described in the previous section (see Algorithm 1 in page 50) to get the persistence intervals. The interval who has death equal to zero means that never died.

```
1000
function matrix = complex2matrix(complex)
1002
s = cellfun(@(x) size(x,2),complex);
1004 dim = unique(s);
matrix = ones(size(s,2),3)*-1;
1006
for i=1:size(complex,2)
1008  if(s(i)==1)
      matrix(i,1) = complex{i};
1010  end
  v =  complex{i};
1012  f=0;
  if(size(v,2)==2)
1014   f = (~isempty(find(v==1)) && ~isempty(find(v==37))); end
  if(s(i)==2)
1016      v =  complex{i};
      in1 = 0;inn2=0;
1018      con = 0;
      for j=i-1:-1:1
1020        if(size(complex{j},2)==1 && v(1,1)==complex{j} )
              matrix(i,1) = j-1;
1022          con = con + 1;
        end
1024        if(size(complex{j},2)==1 && v(1,2)==complex{j} )
              matrix(i,2) = j-1;
1026          con = con + 1;
        end
1028        if(con==2)
              con = 0;
1030          break;
```

```
        end
1032
      end
1034  end

1036  if(s(i)==3)
      v =  complex{i};
1038  in1 = 0;inn2=0;
      con = 0;
1040  for j=i-1:-1:1
        if(size(complex{j},2)==2 && ( ~isempty(find(v(1,1)==complex{j
    })) && ~isempty(find(v(1,2)==complex{j})) ))
1042          matrix(i,1) = j-1;
              con = con + 1;
1044      end
        if(size(complex{j},2)==2 && ( ~isempty(find(v(1,1)==complex{j
    })) && ~isempty(find(v(1,3)==complex{j})) ) )
1046          matrix(i,2) = j-1;
              con = con + 1;
1048      end
        if(size(complex{j},2)==2 && ( ~isempty(find(v(1,2)==complex{j
    })) && ~isempty(find(v(1,3)==complex{j})) ) )
1050          matrix(i,3) = j-1;
              con = con + 1;
1052      end
        if(con==3)
1054          con = 0;
              break;
1056      end

1058  end
  end
1060 end

1062 end
```

The next code for our algorithm gets the persistence intervals, and later calculates the persistent entropy based on these intervals.

```
1000 matrix = complex2matrix(Complejo{k});

1002          [cc]= Persistence_new(matrix);
              cc=sortrows(cc,1);
1004          all=1:length(matrix);
```

```
            missing=setdiff(all,cc(:,1))';
            missing(:,2)=0;
            dd= sortrows([cc;missing],1);
            ii= Index{k};
            ii=[0,ii]';
            for h=1:length(dd)
                dd(h,1)=max(cumsum((ii-dd(h,1)<0)));
                dd(h,2)=max(cumsum((ii-dd(h,2)<0)));
            end
                dd(dd(:, 2)== 0, :)= [];
                dd(dd(:, 1)== dd(:, 2), :)= [];
             [entropy] = per_entropy(dd);
             diagram{k}=dd;
             ent{k}=entropy;
```

To classify, we used the APP Classification Learner which offers Matlab. Thanks to that, it is possible to explore supervised machine learning using various classifiers. The automated training allows us to find the best classification model, includes decision trees, discriminant analysis, support vector machines, logistic regression, nearest neighbors, naive Bayes, and ensemble classification.

The model uses a training dataset to train it and, later, it uses a validation dataset to test it and get the accuracy of the model. After we export the model to the workspace as trained_model (variable), we can predict the model with any sample.

```

model= trainedModel.predictFcn(Sample);

%dataset
T = cvpartition(Entropy\_persistence.label,'HoldOut',0.2); %20% use
    for testing data
triidx = training(T);
testing = dataset(~triidx,:);
training = Entropy\_persistence(triidx,:);
%Use classification learning apps
%Select training to train and validate
%Export Model
%Classify the testing dataset using trainedModel
model = trainedModel.predictFcn(testing);
% Perform accuracy using for example,loss, accuracy, confusion..
```

6.2 Analysis of the implementation of the algoritm for the audio dataset

Considering the context of the project and the existing software environment, the language Python was chosen as the main programming language to implement the second part. This is a trade-off about to find a balance between the ease of implementation due to the language's machine learning libraries, and the ease of use for future developers.

Regarding the libraries, there are some used to work with audio signals. The first one is the *Librosa* library [80], whose overall goal is to provide a set of functions necessary to create audio information retrieval systems. Preprocessing of the audio signal was done using *Librosa*. Another useful library applied is *pandas* [81] that provides ways to store and organize data-sets efficiently.

Based on the paradigm of *scikit-learn library*, the giotto-tda library arises to simplify the principal items for topological machine learning. The library currently supports the application of persistent homology to graph and time-series data.

To calculate persistent homology, it is necessary to instantiate a Vietoris-Rips Persistence transformer and calculate the persistence diagram for the collection of point clouds. The topological features can be a connected component, 1D hole/loop, 2D cavity, or more generally n-dimensional *void* that exists in the data at scales between its birth and death values. The homology dimension of the feature is stored as the third input in that triplet.

```
1000
from gtda.homology import VietorisRipsPersistence
1002 VR = VietorisRipsPersistence(homology_dimensions=[0, 1, 2])  #
    Parameter explained in the text
diagrams = VR.fit_transform(point_clouds)
1004 diagrams.shape
```

Listing 6.2: VietorisRipsPersistence

As we explained in the thesis, the Embedding Theorem is practical when we have a time-series, which is expected to be guided by a non-linear system. The embedding dimension d and time delay τ can be choose manually. However, there are two techniques that can be applied to get these parameter automatically. Mutual Information to calculate τ and False nearest neighbours to get d. The next two function show how

this thesis implement both algorithms to calculate the optimal Embedding parameters in the dataset used.
The function *takensEmbedding* returns the Takens embedding of data with delay into dimension, delay * dimension must be < len(data).

```
def takensEmbedding(data, delay, dimension):
    if delay*dimension > len(data):
        raise NameError('Delay times dimension exceed length of data!')
    embeddedData = np.array([data[0:len(data)-delay*dimension]])
    for i in range(1, dimension):
        embeddedData = np.append(embeddedData, [data[i*delay:len(data) - delay*(dimension - i)]], axis=0)
    return embeddedData
```

Then, there are two measures which give the idea of which delay and dimension parameters to chose. For the first one, mutual information is applied as follows.

```

def mutualInformation(data, delay, nBins):
    #"This function gets the mutual information"
    I = 0
    xmax = max(data)
    xmin = min(data)
    delayData = data[delay:len(data)]
    shortData = data[0:len(data) - delay]
    sizeBin = abs(xmax - xmin) / nBins

    probInBin = {}
    conditionBin = {}
    conditionDelayBin = {}
    for h in range(0, nBins):
        if h not in probInBin:
            conditionBin.update({h: (shortData >= (xmin + h * sizeBin)) & (shortData < (xmin + (h + 1) * sizeBin))})
            probInBin.update({h: len(shortData[conditionBin[h]]) / len(shortData)})
        for k in range(0, nBins):
            if k not in probInBin:
                conditionBin.update(
                    {k: (shortData >= (xmin + k * sizeBin)) & (shortData < (xmin + (k + 1) * sizeBin))})
                probInBin.update({k: len(shortData[conditionBin[k]]) / len(shortData)})
```

```
            if k not in conditionDelayBin:
                conditionDelayBin.update(
                    {k: (delayData >= (xmin + k * sizeBin)) & (
delayData < (xmin + (k + 1) * sizeBin))})
            Phk = len(shortData[conditionBin[h] & conditionDelayBin[
k]]) / len(shortData)
            if Phk != 0 and probInBin[h] != 0 and probInBin[k] != 0:
                I -= Phk * math.log(Phk / (probInBin[h] * probInBin[
k]))
    return I
```

After, we got the mutual information for each audio signal in dependence of τ, the algorithm choose the first minimum value into range of 21 moments. The number of bins is choosen to be 10 since the audio signal has ~1000 data points, then we expect to have around 100 points in each bin.

```
datDelayInformation = []
  for i in range(1, 21):
      try:
          datDelayInformation = np.append(datDelayInformation, [
mutualInformation(Signal, i, 10)])
      except:
          pass
  plt.plot(range(1, 21), datDelayInformation)
  plt.xlabel('delay')
  plt.ylabel('mutual information')
```

Persim is a Python library that includes many tools to analyze the persistence diagrams obtained. From this library, we used the next function in order to perform the persistent entropy values of the persistence diagrams, assuming that the diagrams are finite.

```

persistent_entropy(dgms[, keep_inf,...])
```

The Gudhi library (Geometry Understanding in Higher Dimensions) [76] is a Python module for Computational Topology and Topological Data Analysis. It provides easy and efficient implementations of algorithms and functions of this area. We used the bottleneck distance between two persistence diagram files.

The function *gudhi.plot_persistence_diagram* allows to plot the persistence diagram from persistence values list providing a np.array (of dimension $N \times 2$) representing a diagram in a single homology dimension.

Scikit-learn library [97] is used in this context for some utilities such as splitting a dataset into train validation and test sets. The pre-processing module provides a utility class *StandardScaler* that computes the mean and standard deviation on a training set to be able to later reapply the same transformation on the testing set. We use some classifiers offered by this library: specifically, the class *KNeighborsClassifier* that implemented the k-nearest neighbors' vote where, by default, it uses the distance metric Minkowski, but there is a list of available metrics.

In the third experiment, in order to compute the input for the classification, the metric was "precomputed" defining $\mathbb{X}$ as a square distance matrix with the bottleneck distance between persistence diagrams.

The *Sklearn* library implements estimators to solve classification problems of kind multiclass or multilabel by decomposing such issues into a binary classification problem. To mix the classification in video signals and audio signals, two different strategies used in this approach were One-vs-Rest and One-vs-One. They have been taken from this library.

The next example shows, the classifier *OneVsRes* with a Logisitic Regression class implemented as the binary classification model.

```
# logistic regression for multi-class classification using a one-vs-rest
from sklearn.datasets import make_classification
from sklearn.linear_model import LogisticRegression
from sklearn.multiclass import OneVsRestClassifier
# define dataset
X, y = make_classification(n_samples=141, n_features=9, n_informative=3, n_redundant=3, n_classes=7, random_state=1)
# define model
model = LogisticRegression()
# define the ovr strategy
ovr = OneVsRestClassifier(model)
# fit model
ovr.fit(X, y)
# make predictions
yhat = ovr.predict(X)
```

Similar to previous one, the library provides the OneVsOne strategy to be used with any classifier.

The function *classification_report* offers a text report with the main classification metrics. Here, we see an example of one classification from the thesis that we explained in the previous chapter.

```
[[84 10 10 13 13  8  9]
 [25 69 11 12 10 10  7]
 [19 23 68 13 11 11 11]
 [20 23 22 64 15 10 12]
 [20 20 13 17 72 11  3]
 [29 23 20 15  6 58  6]
 [27 17 20 14 16 17 38]]
```

The function *confusion_matrix* computes the confusion matrix to evaluate the accuracy of a classification with each row corresponding to the true class. By definition, entry (p, q) in the matrix is the number of observations belongs to group p, but predicted to be in q. Here is an example of the accuracy output of the algorithm.

```
              precision    recall  f1-score   support

         2.0       0.38      0.57      0.45       147
         3.0       0.37      0.48      0.42       144
         4.0       0.41      0.44      0.43       156
         5.0       0.43      0.39      0.41       166
         6.0       0.50      0.46      0.48       156
         7.0       0.46      0.37      0.41       157
         8.0       0.44      0.26      0.32       149

    accuracy                           0.42      1075
   macro avg       0.43      0.42      0.42      1075
weighted avg       0.43      0.42      0.42      1075
```

7 | Conclusions and future works

In this thesis, we have developed an application of the persistent entropy to extract information from the videos and solve a classification problem for the emotions described.

Regarding the video-only emotion recognition method developded in this thesis, we can conclude that the application of the eight filtrations in different directions to the landmark points of the face allows us to confirm that each filtration contains relevant information for classifying. The results reached with the combination of each filtration yielded promising and competitive results.

Regarding the audio-signal emotion recognition approach proposed in this thesis, we have to say that since the Embedding Theorem is very useful in situations where one has a time-series, our initial idea was to apply it to the audio signal. Later, we constructed the complex on this embedding signal and calculate the persistent entropy value expecting to have good results. Nevertheless, the persistent entropy did not achieve a good classification. According to the theorem, these time series are expected to be driven by a non-linear system and should be guaranteed that the resulting embedding is a faithful reproduction of the original dynamics. In our case, that assertion did not seems to happen, one reason is given by these time series are not homogeneous or cyclical.

Finally, based on the previous results given in [44] where a topological model was applied to obtain a single real number from each raw signal using persistent entropy, and the successful approach obtained in this thesis using video-signal only, we have introduced here a new method for emotion recognition combining audio and video signal obtaining very good results compared to state-of-the-art methods.

Specifically, we have followed several ideas to work where we have reached promising results. The second and third experiments in the audio dataset section evidence

it. The idea was similar to the previous work of our group but, in this case, we use a matrix distance based on the bottleneck distance of the persistence diagram as the input of the classification algorithm. This way, the accuracy increases. Nevertheless, we think that more research on audio emotional recognition task can be done. The inclusion of other types of information that the audio offers, mix them, or the use of an auxiliary neural network to achieve high-level features should improve the accuracy of our model.

Furthermore, to get better results in the classification of emotion task, we have combined the topological information obtained from the video dataset with the topological information obtained from the audio dataset. The setting One-vs-Rest allowed to achieve competitive results and be in concordance with one of the most relevant conclusions of the KRISTINA project: "The combination of visual and audio features can develop better predictions than using them separately".

The following future works are plan to be explored:

- To Expand to a n-dimensional feature vector of raw audio signals by extracting other types of information.
- To divide the landmark points into different subsets to determine regions or pairs of regions that contain discriminative landmark points for each facial expression.
- To reduce dimension could be an important task in the future since introducing new subsets of landmark points, the dimension of the feature vector could increase.
- To analyze 3D data of the video sequence. The RAVDESS dataset provides landmarks points in the 3-dimensional space $\mathbb{R}^3$. Taking advantage of the depth information of the landmark points could be a challenging problem for the future.
- Instead of only focus on the face, another interesting point is to consider the silhouettes of the people that appear in the videos which include the face and the shoulder movements and that could have relevant information for the emotion recognition purpose.
- The temporal relation between audio and visual features is not well-explored in the literature. In a future study, we will try to integrate this relation into our proposed system.

Bibliography

[1] Albano, A., Passamante, A., and Farrell, M. E. Using higher-order correlations to define an embedding window. *Physica D: Nonlinear Phenomena 54*, 1-2 (1991), 85–97.

[2] Alswaidan, N., and Menai, M. E. B. A survey of state-of-the-art approaches for emotion recognition in text. *Knowledge and Information Systems* (2020), 1–51.

[3] Atienza, N., Escudero, L. M., Jimenez, M. J., and Soriano-Trigueros, M. Persistent entropy: a scale-invariant topological statistic for analyzing cell arrangements. *arXiv preprint arXiv:1902.06467* (2019).

[4] Atienza, N., González-Díaz, R., and Soriano-Trigueros, M. On the stability of persistent entropy and new summary functions for tda. *arXiv preprint arXiv:1803.08304* (2018).

[5] Avots, E., Sapiński, T., Bachmann, M., and Kamińska, D. Audiovisual emotion recognition in wild. *Machine Vision and Applications 30*, 5 (2019), 975–985.

[6] Bakker, I., Van der Voordt, T., Vink, P., and de Boon, J. Pleasure, arousal, dominance: Mehrabian and russell revisited. *Current Psychology 33*, 3 (2014), 405–421.

[7] Bergomi, M. G., and Baratè, A. Homological persistence in time series: an application to music classification. *Journal of Mathematics and Music* (2020), 1–18.

[8] Binchi, J., Merelli, E., Rucco, M., Petri, G., and Vaccarino, F. jholes: A tool for understanding biological complex networks via clique weight rank persistent homology. *Electron. Notes Theor. Comput. Sci. 306* (2014), 5–18.

[9] Borsuk, K. On the imbedding of systems of compacta in simplicial complexes. *Fundamenta Mathematicae 35*, 1 (1948), 217–234.

[10] Botnan, M. B. Three approaches in computational geometry and topology: Persistent homology, discrete differential geometry and discrete morse theory. Master's thesis, Institutt for matematiske fag, 2011.

[11] Buzug, T., and Pfister, G. Comparison of algorithms calculating optimal embedding parameters for delay time coordinates. *Physica D: Nonlinear Phenomena 58*, 1-4 (1992), 127–137.

[12] Buzug, T., and Pfister, G. Optimal delay time and embedding dimension for delay-time coordinates by analysis of the global static and local dynamical behavior of strange attractors. *Physical review A 45*, 10 (1992), 7073.

[13] Calò, D. G. Gaussian mixture model classification: A projection pursuit approach. *Computational statistics & data analysis 52*, 1 (2007), 471–482.

[14] Camplani, M., and Cannas, B. The role of the embedding dimension and time delay in time series forecasting. *IFAC Proceedings Volumes 42*, 7 (2009), 316–320.

[15] Chazal, F., Cohen-Steiner, D., Guibas, L. J., Mémoli, F., and Oudot, S. Y. Gromov-hausdorff stable signatures for shapes using persistence. In *Computer Graphics Forum* (2009), vol. 28, Wiley Online Library, pp. 1393–1403.

[16] Chazal, F., De Silva, V., Glisse, M., and Oudot, S. The structure and stability of persistence modules. *arXiv preprint arXiv:1207.3674 21* (2012).

[17] Chazal, F., De Silva, V., and Oudot, S. Persistence stability for geometric complexes. *Geometriae Dedicata 173*, 1 (2014), 193–214.

[18] Chazal, F., Fasy, B. T., Lecci, F., Rinaldo, A., Singh, A., and Wasserman, L. On the bootstrap for persistence diagrams and landscapes. *arXiv preprint arXiv:1311.0376* (2013).

[19] Cherniakov, M. *An introduction to parametric digital filters and oscillators*. Wiley Online Library, 2003.

[20] Chintakunta, H., Gentimis, T., Gonzalez-Diaz, R., Jimenez, M.-J., and Krim, H. An entropy-based persistence barcode. *Pattern Recognition 48*, 2 (2015), 391 – 401.

[21] Cohen-Steiner, D., Edelsbrunner, H., and Harer, J. Stability of persistence diagrams. *Discrete & computational geometry 37*, 1 (2007), 103–120.

[22] Cohen-Steiner, D., Edelsbrunner, H., Harer, J., and Mileyko, Y. Lipschitz functions have l p-stable persistence. *Foundations of computational mathematics 10*, 2 (2010), 127–139.

[23] Cohen-Steiner, D., Edelsbrunner, H., and Morozov, D. Vines and vineyards by updating persistence in linear time. In *Proceedings of the twenty-second annual symposium on Computational geometry* (2006), pp. 119–126.

[24] Cunningham, P., and Delany, S. J. k-nearest neighbour classifiers–. *arXiv preprint arXiv:2004.04523* (2020).

[25] Dalgleish, T., and Power, M. *Handbook of cognition and emotion*. John Wiley & Sons, 2000.

[26] De Silva, V., and Ghrist, R. Coverage in sensor networks via persistent homology. *Algebraic & Geometric Topology 7*, 1 (2007), 339–358.

[27] Devi, J. S., Yarramalle, S., and Nandyala, S. P. Speaker emotion recognition based on speech features and classification techniques. *International Journal of Image, Graphics and Signal Processing 6*, 7 (2014), 61.

[28] Diener, E., Larsen, R. J., Levine, S., and Emmons, R. A. Intensity and frequency: dimensions underlying positive and negative affect. *Journal of personality and social psychology 48*, 5 (1985), 1253.

[29] d'Amico, M., Frosini, P., and Landi, C. Natural pseudo-distance and optimal matching between reduced size functions. *Acta applicandae mathematicae 109*, 2 (2010), 527–554.

[30] Edelsbrunner, H., and Harer, J. *Computational topology: an introduction*. American Mathematical Soc., 2010.

[31] Edelsbrunner, H., and Mücke, E. P. Three-dimensional alpha shapes. *ACM Transactions on Graphics (TOG) 13*, 1 (1994), 43–72.

[32] El Ayadi, M., Kamel, M. S., and Karray, F. Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition 44*, 3 (2011), 572–587.

[33] Fasy, B. T., Lecci, F., Rinaldo, A., Wasserman, L., Balakrishnan, S., Singh, A., et al. Confidence sets for persistence diagrams. *The Annals of Statistics 42*, 6 (2014), 2301–2339.

[34] France, D. J., Shiavi, R. G., Silverman, S., Silverman, M., and Wilkes, M. Acoustical properties of speech as indicators of depression and suicidal risk. *IEEE transactions on Biomedical Engineering 47*, 7 (2000), 829–837.

[35] Frank, J., Mannor, S., and Precup, D. Activity and gait recognition with time-delay embeddings. In *AAAI* (2010), vol. 2, Citeseer, p. 102.

[36] Fraser, A. M. Information and entropy in strange attractors. *IEEE transactions on Information Theory 35*, 2 (1989), 245–262.

[37] Garcia Cifuentes, C. *Multi-class Classification with Machine Learning and Fusion.* Universitat Politècnica de Catalunya, 2009.

[38] Gery, I., Miljkovitch, R., Berthoz, S., and Soussignan, R. Empathy and recognition of facial expressions of emotion in sex offenders, non-sex offenders and normal controls. *Psychiatry research 165*, 3 (2009), 252–262.

[39] Ghimire, D., and Lee, J. Geometric feature-based facial expression recognition in image sequences using multi-class adaboost and support vector machines. *Sensors 13*, 6 (2013), 7714–7734.

[40] Ghrist, R. Barcodes: the persistent topology of data. *Bulletin of the American Mathematical Society 45*, 1 (2008), 61–75.

[41] Gidea, M., and Katz, Y. Topological data analysis of financial time series: Landscapes of crashes. *Physica A: Statistical Mechanics and its Applications 491* (2018), 820–834.

[42] Giesen, J., and John, M. The flow complex: a data structure for geometric modeling. *Computational Geometry 39*, 3 (2008), 178–190.

[43] Gonzalez-Diaz, R., Jimenez, M.-J., and Krim, H. Towards minimal barcodes. In *International Workshop on Graph-Based Representations in Pattern Recognition* (2013), Springer, pp. 184–193.

[44] Gonzalez-Diaz, R., Paluzo-Hidalgo, E., and Quesada, J. F. Towards emotion recognition: A persistent entropy application. In *Computational Topology in Image Context* (2019), R. Marfil, M. Calderón, F. Díaz del Río, P. Real, and A. Bandera, Eds., Springer International Publishing, pp. 96–109.

[45] Gonzalez-Diaz, R., and Real, P. On the cohomology of 3d digital images. *Discrete Applied Mathematics 147*, 2-3 (2005), 245–263.

[46] Grandjean, D., Sander, D., and Scherer, K. R. Conscious emotional experience emerges as a function of multilevel, appraisal-driven response synchronization. *Consciousness and cognition 17*, 2 (2008), 484–495.

[47] Griffiths, P. E. Emotions. *A Companion to cognitive science* (2017), 197–203.

[48] Gunes, H., and Pantic, M. Automatic, dimensional and continuous emotion recognition. *International Journal of Synthetic Emotions (IJSE) 1*, 1 (2010), 68–99.

[49] Guo, X., Polanía, L. F., and Barner, K. E. Audio-video emotion recognition in the wild using deep hybrid networks. *arXiv preprint arXiv:2002.09023* (2020).

[50] Hatcher, A. *Algebraic topology*. cambridge university press, 2002.

[51] Hegger, R., Kantz, H., and Schreiber, T. Practical implementation of nonlinear time series methods: The tisean package. *Chaos: An Interdisciplinary Journal of Nonlinear Science 9*, 2 (1999), 413–435.

[52] Hilton, P. J., and Wylie, S. *Homology theory: An introduction to algebraic topology*. CUP Archive, 1967.

[53] Hozjan, V., and Kačič, Z. Context-independent multilingual emotion recognition from speech signals. *International journal of speech technology 6*, 3 (2003), 311–320.

[54] Hu, H., Xu, M.-X., and Wu, W. Fusion of global statistical and segmental spectral features for speech emotion recognition. In *Eighth Annual Conference of the International Speech Communication Association* (2007).

[55] Issa, D., Demirci, M. F., and Yazici, A. Speech emotion recognition with deep convolutional neural networks. *Biomedical Signal Processing and Control 59* (2020), 101894.

[56] Jain, V., Lamba, P. S., Singh, B., Namboothiri, N., and Dhall, S. Facial expression recognition using feature level fusion. *Journal of Discrete Mathematical Sciences and Cryptography 22*, 2 (2019), 337–350.

[57] Jenke, R., Peer, A., and Buss, M. Feature extraction and selection for emotion recognition from eeg. *IEEE Transactions on Affective computing 5*, 3 (2014), 327–339.

[58] JIAYU, L., YUEKE, W., ZHIPING, H., AND ZHENKANG, S. Selection of proper time-delay in phase space reconstruction of speech signals [j]. *Signal Processing 3* (1999).

[59] KAMIŃSKA, D., SAPIŃSKI, T., AND PELIKANT, A. Recognition of emotion intensity basing on neutral speech model. In *Man-Machine Interactions 3*. Springer, 2014, pp. 451–458.

[60] KANTZ, H., AND SCHREIBER, T. *Nonlinear time series analysis*, vol. 7. Cambridge university press, 2004.

[61] KENNEL, M. B., BROWN, R., AND ABARBANEL, H. D. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical review A 45*, 6 (1992), 3403.

[62] KEOGH, E., AND KASETTY, S. On the need for time series data mining benchmarks: a survey and empirical demonstration. *Data Mining and knowledge discovery 7*, 4 (2003), 349–371.

[63] KHASAWNEH, F. A., AND MUNCH, E. Chatter detection in turning using persistent homology. *Mechanical Systems and Signal Processing 70* (2016), 527–541.

[64] KIM, Y., LEE, H., AND PROVOST, E. M. Deep learning for robust feature generation in audiovisual emotion recognition. In *2013 IEEE international conference on acoustics, speech and signal processing* (2013), IEEE, pp. 3687–3691.

[65] KLEINSMITH, A., AND BIANCHI-BERTHOUZE, N. Affective body expression perception and recognition: A survey. *IEEE Transactions on Affective Computing 4*, 1 (2012), 15–33.

[66] KOSSAIFI, J., TZIMIROPOULOS, G., TODOROVIC, S., AND PANTIC, M. Afew-va database for valence and arousal estimation in-the-wild. *Image and Vision Computing 65* (2017), 23–36.

[67] KRYZHANOVSKY, B., DUNIN-BARKOWSKI, W., AND REDKO, V. Advances in neural computation, machine learning, and cognitive research. *Neuroinformatics* (2017).

[68] LAMAR-LEON, J., ALONSO-BARYOLO, R., GARCIA-REYES, E., AND GONZALEZ-DIAZ, R. Persistent homology-based gait recognition robust to upper body variations. In *2016 23rd International Conference on Pattern Recognition (ICPR)* (Dec 2016), pp. 1083–1088.

[69] Lamar-León, J., Baryolo, R. A., Reyes, E. B. G., and González-Díaz, R. Persistent-homology-based gait recognition. *CoRR abs/1707.06982* (2017).

[70] Lazar, E. A., Mason, J. K., MacPherson, R. D., and Srolovitz, D. J. Statistical topology of three-dimensional poisson-voronoi cells and cell boundary networks. *Physical Review E 88*, 6 (2013), 063309.

[71] Leinonen, L., Hiltunen, T., Linnankoski, I., and Laakso, M.-L. Expression of emotional–motivational connotations with a one-word utterance. *The Journal of the Acoustical society of America 102*, 3 (1997), 1853–1863.

[72] Li, W., Tsangouri, C., Abtahi, F., and Zhu, Z. A recursive framework for expression recognition: from web images to deep models to game dataset. *Machine Vision and Applications 29*, 3 (2018), 489–502.

[73] Liu, T.-Y., Yang, Y., Wan, H., Zeng, H.-J., Chen, Z., and Ma, W.-Y. Support vector machines classification with a very large-scale taxonomy. *Acm Sigkdd Explorations Newsletter 7*, 1 (2005), 36–43.

[74] Livingstone, S. R., and Russo, F. A. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PloS one 13*, 5 (2018), e0196391.

[75] Ma, J., Jin, H., Yang, L. T., and Tsai, J. J.-P. *Ubiquitous Intelligence and Computing: Third International Conference, UIC 2006, Wuhan, China, September 3-6, 2006, Proceedings (Lecture Notes in Computer Science)*. Springer-Verlag, 2006.

[76] Maria, C., Boissonnat, J.-D., Glisse, M., and Yvinec, M. The gudhi library: Simplicial complexes and persistent homology. In *International Congress on Mathematical Software* (2014), Springer, pp. 167–174.

[77] Marsh, A. A., Kozak, M. N., and Ambady, N. Accurate identification of fear facial expressions predicts prosocial behavior. *Emotion 7*, 2 (2007), 239.

[78] Mayer, J. D., Caruso, D. R., and Salovey, P. The ability model of emotional intelligence: Principles and updates. *Emotion review 8*, 4 (2016), 290–300.

[79] McDuff, D., Kaliouby, R., Senechal, T., Amr, M., Cohn, J., and Picard, R. Affectiva-mit facial expression dataset (am-fed): Naturalistic and spontaneous facial expressions collected. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (2013), pp. 881–888.

[80] McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., and Nieto, O. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference* (2015), vol. 8, pp. 18–25.

[81] McKinney, W., et al. pandas: a foundational python library for data analysis and statistics. *Python for High Performance and Scientific Computing 14*, 9 (2011).

[82] MELLO, R. F., and Ponti, M. A. *Machine Learning: A Practical Approach on the Statistical Learning Theory*. Springer, 2018.

[83] Merelli, E., Piangerelli, M., Rucco, M., and Toller, D. A topological approach for multivariate time series characterization: the epileptic brain. In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)* (2016), pp. 201–204.

[84] Mileyko, Y., Mukherjee, S., and Harer, J. Probability measures on the space of persistence diagrams. *Inverse Problems 27*, 12 (2011), 124007.

[85] Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of machine learning*. MIT press, 2018.

[86] Morozov, D. *Homological illusions of persistence and stability*. Duke University, 2008.

[87] Moskvin, A., and Shishkin, A. Deep learning based human emotional state recognition in a video. *Journal of Modeling and Optimization 12*, 1 (2020), 51–59.

[88] Muller, K.-R., Mika, S., Ratsch, G., Tsuda, K., and Scholkopf, B. An introduction to kernel-based learning algorithms. *IEEE transactions on neural networks 12*, 2 (2001), 181–201.

[89] Munch, E., Turner, K., Bendich, P., Mukherjee, S., Mattingly, J., Harer, J., et al. Probabilistic fréchet means for time varying persistence diagrams. *Electronic Journal of Statistics 9*, 1 (2015), 1173–1204.

[90] Nicole, J., Rapp, V., Bailly, K., Prevost, L., and Chetouani, M. Audio-visual emotion recognition: A dynamic, multimodal approach. *Preprint-HAL archives* (2014).

[91] Noroozi, F., Kaminska, D., Corneanu, C., Sapinski, T., Escalera, S., and Anbarjafari, G. Survey on emotional body gesture recognition. *IEEE transactions on affective computing* (2018).

[92] Ofodile, I., Kulkarni, K., Corneanu, C. A., Escalera, S., Baro, X., Hyniewska, S., Allik, J., and Anbarjafari, G. Automatic recognition of deceptive facial expressions of emotion. *arXiv preprint arXiv:1707.04061 2* (2017).

[93] Otter, N., Porter, M. A., Tillmann, U., Grindrod, P., and Harrington, H. A. A roadmap for the computation of persistent homology. *EPJ Data Science 6*, 1 (2017), 17.

[94] Oudre, L., Jakubowicz, J., Bianchi, P., and Simon, C. Classification of periodic activities using the wasserstein distance. *IEEE transactions on biomedical engineering 59*, 6 (2012), 1610–1619.

[95] Owada, T., Adler, R. J., et al. Limit theorems for point processes under geometric constraints (and topological crackle). *The Annals of Probability 45*, 3 (2017), 2004–2055.

[96] Pan, S., and Duraisamy, K. Data-driven discovery of closure models. *SIAM Journal on Applied Dynamical Systems 17*, 4 (2018), 2381–2413.

[97] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12* (2011), 2825–2830.

[98] Penrose, M., et al. *Random geometric graphs*, vol. 5. Oxford university press, 2003.

[99] Pereira, C. M., and de Mello, R. F. Persistent homology for time series and spatial data clustering. *Expert Systems with Applications 42*, 15-16 (2015), 6026–6038.

[100] Pławiak, P., Sośnicki, T., Niedźwiecki, M., Tabor, Z., and Rzecki, K. Hand body language gesture recognition based on signals from specialized glove and machine learning algorithms. *IEEE Transactions on Industrial Informatics 12*, 3 (2016), 1104–1113.

[101] Plutchik, R. The nature of emotions: Human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice. *American scientist 89*, 4 (2001), 344–350.

[102] Poria, S., Cambria, E., Bajpai, R., and Hussain, A. A review of affective computing: From unimodal analysis to multimodal fusion. *Information Fusion 37* (2017), 98–125.

[103] Robinson, A., and Turner, K. Hypothesis testing for topological data analysis. *arXiv preprint arXiv:1310.7467* (2013).

[104] Rosenstein, M. T., Collins, J. J., De Luca, C. J., et al. Reconstruction expansion as a geometry-based framework for choosing proper delay times. *Physica-Section D 73*, 1 (1994), 82–98.

[105] Rucco, M., Gonzalez-Diaz, R., Jimenez, M.-J., Atienza, N., Cristalli, C., Concettoni, E., Ferrante, A., and Merelli, E. A new topological entropy-based approach for measuring similarities among piecewise linear functions. *Signal Processing 134* (2017), 130–138.

[106] Sapiński, T., Kamińska, D., Pelikant, A., Ozcinar, C., Avots, E., and Anbarjafari, G. Multimodal database of emotional speech, video and gestures. In *International Conference on Pattern Recognition* (2018), Springer, pp. 153–163.

[107] Schlosberg, H. Three dimensions of emotion. *Psychological review 61*, 2 (1954), 81.

[108] Schuller, B., and Batliner, A. *Computational Paralinguistics: Emotion, Affect and Personality in Speech and Language Processing*. Wiley, 2013.

[109] Schuller, B., Rigoll, G., and Lang, M. Speech emotion recognition combining acoustic features and linguistic information in a hybrid support vector machine-belief network architecture. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing* (2004), vol. 1, IEEE, pp. I–577.

[110] Senin, P. Dynamic time warping algorithm review. *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA 855*, 1-23 (2008), 40.

[111] Seversky, L. M., Davis, S., and Berger, M. On time-series topological data analysis: New data and opportunities. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (2016), pp. 59–67.

[112] Shahin, I., and Nassif, A. B. Three-stage speaker verification architecture in emotional talking environments. *International Journal of Speech Technology 21*, 4 (2018), 915–930.

[113] Shalev-Shwartz, S., and Ben-David, S. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[114] Shami, M. T., and Kamel, M. S. Segment-based approach to the recognition of emotions in speech. In *2005 IEEE International Conference on Multimedia and Expo* (2005), IEEE, pp. 4–pp.

[115] Shannon, C. E. A mathematical theory of communication. *The Bell system technical journal 27*, 3 (1948), 379–423.

[116] Sheehy, D. R. Linear-size approximations to the vietoris–rips filtration. *Discrete & Computational Geometry 49*, 4 (2013), 778–796.

[117] Shojaeilangari, S., Yau, W.-Y., and Teoh, E.-K. Pose-invariant descriptor for facial emotion recognition. *Machine Vision and Applications 27*, 7 (2016), 1063–1070.

[118] Siddiqi, M. H., Ali, R., Khan, A. M., Park, Y.-T., and Lee, S. Human facial expression recognition using stepwise linear discriminant analysis and hidden conditional random fields. *IEEE Transactions on Image Processing 24*, 4 (2015), 1386–1398.

[119] Takens, F. Detecting strange attractors in turbulence. In *Dynamical systems and turbulence, Warwick 1980*. Springer, 1981, pp. 366–381.

[120] Toth, C. D., O'Rourke, J., and Goodman, J. E. *Handbook of discrete and computational geometry*. CRC press, 2017.

[121] Turner, K. Medians of populations of persistence diagrams. *arXiv preprint arXiv:1307.8300* (2013).

[122] Turner, K., Mileyko, Y., Mukherjee, S., and Harer, J. Fréchet means for distributions of persistence diagrams. *Discrete & Computational Geometry 52*, 1 (2014), 44–70.

[123] Umeda, Y. Time series classification via topological data analysis. *Information and Media Technologies 12* (2017), 228–239.

[124] Van Kreveld, M., Schwarzkopf, O., de Berg, M., and Overmars, M. *Computational geometry algorithms and applications*. Springer, 2000.

[125] Ververidis, D., and Kotropoulos, C. Emotional speech recognition: Resources, features, and methods. *Speech communication 48*, 9 (2006), 1162–1181.

[126] VIETORIS, L. Über den höheren zusammenhang kompakter räume und eine klasse von zusammenhangstreuen abbildungen. *Mathematische Annalen 97*, 1 (1927), 454–472.

[127] VINH, N. X., EPPS, J., AND BAILEY, J. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research 11* (2010), 2837–2854.

[128] WAN, J., ESCALERA, S., ANBARJAFARI, G., JAIR ESCALANTE, H., BARÓ, X., GUYON, I., MADADI, M., ALLIK, J., GORBOVA, J., LIN, C., ET AL. Results and analysis of chalearn lap multi-modal isolated and continuous gesture recognition, and real versus fake expressed emotions challenges. In *Proceedings of the IEEE International Conference on Computer Vision Workshops* (2017), pp. 3189–3197.

[129] WANG, X., SOHEL, F., BENNAMOUN, M., GUO, Y., AND LEI, H. Scale space clustering evolution for salient region detection on 3d deformable shapes. *Pattern Recognition 71* (2017), 414–427.

[130] WANG, Y.-Q. An analysis of the viola-jones face detection algorithm. *Image Processing On Line 4* (2014), 128–148.

[131] YANG, B., AND LUGGER, M. Emotion recognition from speech signals using new harmony features. *Signal Processing 90*, 5 (2010), 1415 – 1423. Special Section on Statistical Signal & Array Processing.

[132] ZHANG, B., ESSL, G., AND PROVOST, E. M. Recognizing emotion from singing and speaking using shared models. In *2015 international conference on affective computing and intelligent interaction (acii)* (2015), IEEE, pp. 139–145.

[133] ZHANG, G. P. Neural networks for classification: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 30*, 4 (2000), 451–462.

[134] ZHAO, J., MAO, X., AND CHEN, L. Speech emotion recognition using deep 1d & 2d cnn lstm networks. *Biomedical Signal Processing and Control 47* (2019), 312–323.

[135] ZOMORODIAN, A., AND CARLSSON, G. Computing persistent homology. *Discrete and Computational Geometry 33*, 2 (2005), 249–274.