POLITECNICO DI MILANO
DEPARTMENT OF MATHEMATICS
DOCTORAL PROGRAM IN
MATHEMATICAL MODELS AND METHODS IN ENGINEERING

# GEOMETRIC DATA ANALYSIS: BETWEEN EQUIVALENCE CLASSES AND NON-EUCLIDEAN SPACES.

Doctoral Dissertation of:
**Matteo Pegoraro**

Supervisor:
**Prof. Piercesare Secchi**

The Chair of the Doctoral Program and Tutor:
**Prof. Irene Maria Sabadini**

2021 – XXXIII Cycle

# Abstract

Dealing with complex data often presents challenges that can be successfully tackled with the use of some geometrical tools. In this thesis we approach two such challenges: extracting information from data considered up to equivalence classes and doing statistical analysis in non-Euclidean spaces. The first kind of issues is faced with the use of Topological Data Analysis techniques. We contribute to this field with the introduction of a new family of topological summaries which can be visually represented as trees. To analyze sets of such objects, we introduce also a novel metric structure along with an algorithm to compute it. Next, we develop an application of such framework in the field of Functional Data Analysis, working with functions up to homeomorphic reparametrization.

Moving the problem of doing statistical analysis from raw data to a space of representations, for instance topological summaries, immediately poses the challenge of defining statistical tools in spaces which are often non-Euclidean and badly behaved from many perspectives. Nevertheless, we start to formalize a language to work in the space of the topological representations previously defined and, as a first result, we obtain approximations of Frechét Means in those spaces.

Lastly we propose a novel class of methods to work with probability distributions on the real line, with the $2-$Wasserstein metric. This metric space is richer in structure compared to the others considered in this dissertation, but still has a constrained nature which must be properly taken into account.


"*What could he see, that I was missing?*
*It had to be the glasses.*
*So I made myself an identical pair. (Better, though)*
*And, with those on, I looked around.*"

Gipi, La mia vita disegnata male, pg. 141

# Acknowledgments

Every piece of this thesis is born within a precise human and scientific environment - which extends both to the professional and non professional sphere. And this environment ended up affecting, enriching and insipiring, in many unpredictable and precious ways my scientific work.

For these reasons I would like to thank my supervisor Piercesare, who allowed me to freely explore ideas and topics, even when links with his usual research fields where far from obvious. With constant trust and dialogues he greatly enriched my scientific perspectives and this, in turn, allowed me to get the most out of collaborations and discussions with other colleagues. I want to thank Anna Calissano for having introduced me to the field of geometric statistics, which opened up a whole new set of directions in my research. Similarly, I thank Samuele Mongodi for all the applied aspects of differential and Riemannian geometry that I discovered through him. A special thanks goes also to Mario Beraha, who got me into Wasserstein metrics and Optimal transport, as well as being a very fruitful research partner. With him I made a clear experience of how enriching is the daily collaboration between researches with a very diverse mathematical background.

Finally, I thank my family and my friends.

# Contents

CHAPTER 1

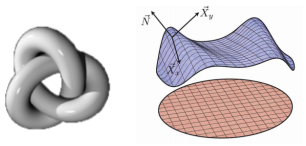
# General Introduction

How do we automatically recognize anomalies in the shape of objects in a production line? Which anomalies should still be accepted and which ones rejected? Are there any relationships between the shape of the brain and neural illnesses? Can we detect the presence of such illnesses at early stages by predicting the evolution of the organ's shape? How do we separate features in a blood vessel which are due to the patient's unique anathomy from features shared among a class of patients? Questions like these are increasingly frequent in data analysis, coming from the most diverse fields, and they all share the problem of facing highly complex data whose variability cannot be naively handled with usual statistical techniques.

Each time new challenges appear, especially in quantitative fields, the first pivotal step towards their solution is the search for the right vocabulary to describe and to talk about the newfound problems. Modeling from scratch the space of statistical units, talking about functions, continuity, distances, curves, deformation of objects, and generalizing concepts like mean and variance, are all aspects of the analysis which need to be carefully and formally addressed. Among the fields of mathematics, the one which is most suited to describe and tackle those fundational challenges is geometry. For this reason the interactions between geometry and data analysis are countless and arising with increasing frequency and potential (Bône, 2020; Bronstein et al., 2021; Curry, 2014; Davis, 2008; Pennec et al., 2019; Tralie, 2017) to the point that terms like "Geometric Statistics", "Geometric Data Analysis" and "Geometric Machine Learning" are

commonly used and attract more and more researchers.

## 1.1 Between Equivalence Classes and non-Euclidean Data



Among the areas of research which interact with the broad concept of geometry, two important research directions can be recognized: the first one is investigating information in geometric objects which is invariant to some kinds of transformations, the second one aims at generalizing vector and differential calculus beyond Euclidean spaces.

Within the first macro area we can find all the different interplays between geometry, topology and algebra. Topological spaces of different nature are often understood in terms of information collected by polynomials and group structures (groups of automorphisms, cohomology groups, groups of bundles etc) and the relationsphips between these algebraic objects and the geometry of the underlying topological spaces are studied in order to grasp how the transformation of a topological space changes the associated groups.

The second main direction is strictly connected with the mathematical questions posed by physics and engineering. Modeling points or quantities bound by certain laws often involves calculation of basic differential quantities like derivatives along some "direction". Problems like studing the change of one quantity with respect to small perturbations of another quantity have always been of main interest in mathematics and physics. However many of such differential problems involve constraints which greatly increase the modeling and calculation complexity. The "curved" and non-Euclidean nature of many constraints asks for non-trivial generalization of operations which are very well understood in linear spaces and go from vector sums and directional derivatives, to the behaviour all kinds of differential operators.

Both research directions provide ideas and results which can help the analyst in dealing with complex data analysis situations of different kinds. The approach which studies the most appropriate mathematical structure whose points are the *atoms* of the statistical analysis, being it the statistical units or a particular representation of such units, goes under the name of Object Oriented Data Analysis (OODA) (Wang and Marron, 2007). This term is intended as a very broad research field which collects the efforts of all the analyses for which the *complexity* of the data set go far beyond "big $n$" or "big $p$" and require non-trivial mathematical modeling of the "sufficient statistics".

This dissertation aims at investigating some constributions that geometry can provide to OODA in both the directions presented in this paragraph.

## 1.2 Data in Equivalence Classes

Transforming variables is one of the most used approaches in statistics. Usually variables are transformed to meet some modellistic hypoteses or to allow a more fruitful comparison or visualization. One of the most used transformations is the standardization of variables, which means measuring them in terms of standard deviations from their mean. In this way the variables are represented with a scale which enables the comparison between phaenomena with different ranges of variability and thus allowing for certain kind of inference and interpretability in the analysis. In other words the statistician is implicitly considering a set of possible representations of such variables and chosing the "optimal" one according to some criterion: among all the possibible scales with which one can express the variables, standardization is selected for its appealing properties. Behind this simple and standard approach there is already the idea of considering not the datum as it is, but a whole equivalence class of possible representations of the same object, among which the practitioner chooses the one that is most suited for the analysis.

This same workflow is applied in many areas of statistics. Consider for instance the case of Functional Data Analysis (FDA). There are many reasons (see Chapter 3 and Ramsay and Silverman (2005) for details) for which functions in a data set may need to be optimally reparametrized: functions are often *aligned* or *registered*, according to some criterion and with parametrization functions belonging to some particular group of transformations. Thus, the real datum is not the single function, but the whole set of possible functions which can be obtained by reparametrizing the "observed" one. Using the language of group actions (Krupka and Saunders, 2011) this set is called the *orbit* of a function under the action of the reparametrization group. From the orbit one optimal representation is usually chosen and, on such representation, the analysis is carried out.

The idea of considering a datum as the orbit of a point under some group action is very powerful and its fruitfully used in many areas of statistics (Eaton, 1989) and machine learning (Bergomi et al., 2019). A different example is given by data augmentation techniques; consider for instance the problem of finding a completely data-driven way to recognize a number, for instance number $9$, whenever it appears in an image. In a completely supervised fashion, a dataset of labelled images is fed to the algorithm to train it for the recognition purpose. In particular, one would like the classification tool to minimize the amount of labelled images required to have good performances on a test set. Clearly the number $9$ can appear in any portion of the image, with any scale and with any perspective. So usually, each labelled image is used to obtain a whole new set of training images for instance by zooming-in in different points, adding noise or applying symmetries, trying to "teach" the algortihm that the number $9$ can appear

in many different forms and places in an image. That is, practitioners aim at including in the training set of an algorithm as many points as possible in the orbit of a certain datum, starting from the ones which are observed, and obtaining part of the remaining ones via different techniques. Ideally one would like the algorithm to "recognize" and "learn" the group action involved, in order to be able to recognize as many instances as possible of the same object.

In some sense the two examples proposed - FDA and data augmentation - provide two different approaches to the same problem: in the case of FDA a convenient representation of the starting object is chosen, while with data augmentation one would like the output of the analysis not to depend on the one representative which is fed to the algorithm, but to be well defined on the whole equivalence class. Where being well defined means that the analysis/pipeline does not distinguish points in the same equivalence class.

The right language to deal with the problem of being well defined up to equivalence relationships is given by quotient spaces, which are literally the sets of equivalence classes of some space under an equivalence relationship. In the case of group actions, the equivalence classes are given by the orbits themselves: each orbit is a single point of the quotient space. Using the language of quotient spaces one can properly define the steps and the tools to be employed in the analysis (Huckemann et al., 2010a) so that the outcome does not depend on a particular representation of the single datum.

A key point which then arises is what kind of information can be extracted from quotient spaces. Topology in this sense, has always be interested in classifying topological spaces considered inside very large equivalence classes. Consider for instance the case of homology and cohomology groups, which are basic topology tools to summarize some topological information of a space (Hatcher, 2000; Munkres, 2018). The information they capture can be interpreted in terms of holes and obstructions and in many cases is easily accessible from the computational point of view, especially if considered with field coefficients. Moreover these groups are invariant to large sets of deformations of the base topological space, induced by homotopy equivalence. These facts make homology an excellent starting points to build tools to extract information from data considered in some quotient space.

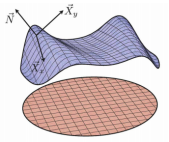
Topological data analysis (TDA) is the name given to a set of techniques which go exactly in this direction: exploiting homology groups to extract information from data. Consider the following examples, concerning two different kinds of data: points clouds and functions. Start with the case of a finite point cloud in $\mathbb{R}^n$. The subspace topology of such object is very poor, and the only information contained in homology groups is the number of connected components, that is the number of points. There are however many ways to build topological spaces starting from a point cloud, which try to capture the "shape" of such point cloud (Chazal and Michel, 2017). It is then quite natural to think that, instead of comparing the information associated to the point clouds, one can compare the homology groups of the topological spaces obtained from the point clouds. Moreover the induced topological spaces are often dependent on one real parameter and

thus one can obtain a whole set of topological spaces parametrized by a subset of the real line. Relationships between spaces obtained at different values of the parameter in many cases allow for a very interesting approach: the homology groups information can in fact be tracked along the ordered family of topological spaces, capturing the most *persistent* homological features that appear in this sequence. This idea is in fact the foundation of *persistent homology* (Edelsbrunner and Harer, 2008). Since the family of induced spaces usually depends only on the pairwise distances between the points in the cloud, applying an isometry to the point cloud does not change the output of the procedure and thus these methods can be used for instance when one wants to consider sets of points up to rigid transformations.

Turn now to the case of a scalar function defined on a topological space. The function can be used to (partially) order the points in the domain: the first points in this ordering are those with lower values, that is the minima, and the last points are the ones where the functions reaches its maxima. To be more precise let $f : X \to \mathbb{R}$. We can induce a sequence of nested topological spaces $X_t := f^{-1}((-\infty, t])$, which is heavily dependent on the values of $f$. Thus observing different functions on the same domain produces different *filtrations* $\{X_t\}_{t\in\mathbb{R}}$. These sequences of topological spaces can be analyzed again by following along $t$ the evolution of the homology groups. Focus on the topological space $X$: if there is some kind of map $\varphi : Y \to X$, one can consider the function $f' : Y \to \mathbb{R}$ given by $f' := f \circ \varphi$. If $\varphi^{-1}$ ensures that the topological information contained in $X_t$ (obtained with $f$), is the same as the topological information contained in $Y_t$ (obtained with $f'$), then the evolution of the homology groups in $\{X_t\}$ and $\{Y_t\}$ is "the same", and thus the functions $f$ and $f'$ are considered as equal by this pipeline. This for instance means that treating functions with this topological approach might find information which is invariant to some kind of reparametrization of the function.

These examples are intended to show that the TDA is a very promising field to design and develop tools which play with the idea of considering data up to certain kind of transformations: the general approach and framework provide a language and a point of view which is a potential fertile ground for new sources of information about data, which present some kind of invariance properties.

## 1.3 Statistics in non-Euclidean Spaces



As highlighted in the previous Section, there are many examples of analyses carried out on data not liying (at least naively) in $\mathbb{R}^n$ or other vector spaces. Quotient spaces of functions up to reparametrizaion, point clouds up to isometries, matrices up to some

kind of base change, numbers up to unit of measure all require complex mathematical frameworks to be treated, since even things like moving from one point to another must be carefully defined and most of the linear structure of the base spaces often becomes useless. The same problem occurs in other situations where data can be seen as embedded in some ambient space but they possess a subspace structure, that is, roughly, that the image of the embedding doesn't naturally fill up the whole space. It is the case for many interesting type of data: symmetric positive definite matrices (like covariance matrices, see Arsigny et al. (2006); Moakher and Zéraï (2011); Pigoli et al. (2014)), sets of orthonormal vectors (James, 1976; Turaga et al., 2011), rotation matrices, densities of probability distributions, even patches of images (Carlsson et al., 2008) cannot be analyzed using the linear structure of the ambient space but must be approached by considering their the structure of *subspace* they belong to. One can then distinguish between two main different situations: the classical situation where one can explicitly describe and model the space or the subspace structure, usually via a set of constraints, but there are also cases in which this structure cannot be modeled a priori, and it must be learned in some way from data, with procedures of *manifold learning* (see for instance Breiding et al. (2018); Budninskiy et al. (2019)). Both the aforementioned situations have attracted a lot of research, but here we focus on the cases in which we have explicit access to the space structure, being it a *manifold*/*submanifold* structure, a *stratified space* structure or other more general kind of spaces.

When the space of interest is determined by a series of constraints inside a linear space, there are two approaches to carry out the statistical analysis: the class of *intrinsic* approaches and the class of *extrinsic* ones. Intrinsic methods (Pennec, 2006) are tools which just rely on the intrinsic structure of the "substructure" the data points belong to (often a *manifold* structure), not on the ambient space, and correctly take into account the real metric structure among data units. The extrinsic approach (Bhattacharya et al., 2012), instead, focuses on how to use the linear space which sorrounds the data points to capture reliable information about data on the subspace. Usually this approach is pursued when intrinsic methods fail to be of any practical use since they are computationally our of reach for the intended purpose. However the intepretability of extrinsic methods is often heavily dependent on the data set: if the variability of the data is small enough, they can often be approximated using a linear subspace of the ambient space (for instance a tangent space), and thus extrinsic methods have a high level of intepretability. When an embedding into a linear space is not avaiable, clearly, the intrinsic approach is the only viable one.

The situation becomes further challenging when one cannot build a differential or even a topological structure which falls into the realm of well-known and deeply studied geometrical objects like manifolds. In this case ad-hoc, meaningful tools and definitions must be carefully obtained in order to be able to work in such spaces (Calissano et al., 2020; Garba et al., 2021; Turner et al., 2014).

### 1.3.1 Probability Distributions

A very interesting class of constrained, non-Euclidean data which arises quite often in data analysis, is the one of probability distributions. There are situations in which either data is given in an aggregated form, or one needs to aggregate data to get reasonable results. For instance it may happen that due to missing information, mistakes, privacy and other reasons, raw data are not avaiable and the analyst only receives summaries of the collected pieces of information. In some other cases modeling the single statistical unit is very complex and one may want to resort some kind of unstructured information which is more readily available, for instance in the case of images in radiomics (Kumar et al., 2012). In these and in many other situations, frequencies of units inside aggregated objects become very important sources of information. Moreover one may want to carry out some statistical analysis to compare situations where the same phaenomenon produces radically different numbers for intrinsic reasons, for instance one may wish to compare frequencies of some events in populations with very different sizes. All these pieces, put together, lead to the analysis of sets of probability distributions.

Probability distributions are constrained objects in that the measure they induce on the whole space is fixed and equal to one. For instance the space of densities of probability measures, is a space of integrable positive functions which integrate to $1$. The non-algebraic nature of these constraints is usually overcome by considering parametrized families of distributions, with either a naive metric on the parameters space, or metrics induced by distances between probability distributions, or even Riemannian metrics, often used in information geometry (Amari, 2021; Ay et al., 2018). There are however some frameworks providing handy representations of big sets of probability distributions, asking for other kinds of assumptions. For instance a set of continuous densities on $\mathbb{R}$ with fixed compact support and satisfying $\int_{\mathbb{R}} log(f) < \infty$, can be given an Hilbert space structure via Bayes spaces (Pawlowsky-Glahn et al., 2014). Another very important case is the one of Wasserstein spaces of probability distributions on $\mathbb{R}$ (Panaretos and Zemel, 2020). The mildness of the assumptions, along the tractability of the spaces' geometric structure, really make Wasserstein spaces a useful tool to work with probability distributions on the real line. On top of that, the Wasserstein metric is also interpretabile, in terms of optimal transport. For all these reasons, a good number of statistical tools have already been defined to work in such spaces.

## 1.4 Outline of the Dissertation

In this thesis we deal with the issues presented in the previous sections. The chapters collect the contributions of the manuscript dividing them between different areas and different research perspectives.

In Chapter 2 and Chapter 3 we present a novel set of topological summaries in the field of TDA, generalizing objects like *merge trees* and *hierarchical dendrograms*. In

particular in Chapter 2 we build a theoretic framework to work with representations of data whose invariance properties are different from the ones of the most widely used TDA tool: *persistence diagrams*. In the same chapter we also develop the computational tools which allow the evaluation of the defined metric in all the examples and case studies carried out in the manuscript. In Chapter 3, instead, we test both a particular instance of the summaries previously defined (merge trees) and persistence diagrams on a benchmark case study in FDA, in order to showcase the effectiveness of the topological data analysis approach in OODA: in this case study, considering functions up to some preparametrization group is almost mandatory. In Chapter 4 we explore the - non-Euclidean - structure of the metric spaces defined in Chapter 2, considering the particular case of merge trees. We start to investigate its topological and metric properties with an ad-hoc definition of the "tangent bundle", paying particular attention to *Frechét means*, which are objects of great interest in data analysis. We end up this part of the thesis with Chapter 5 in which we discuss possible further developments of the research topics presented in the first part of the thesis. In Chapter 6, we change topic and tackle the problem of developing statistical methods to work in the 2-Wasserstein space of probability measures on $\mathbb{R}$. We conclude the dissertation drawing some general conclusions and suggesting some other further research directions in Chapter 7. In the last chapter, Chapter 8, we describe the implementations and the code needed to run all the simulations an analyses of the dissertation.

## 1.5 Note to the reader

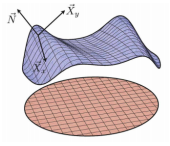
As already stated, this PhD thesis investigates two kinds of interactions between data analysis and geometry: first employing representations of data based on some kind of topological information, second trying to generalize objects which are well defined and understood in usual linear spaces. Moreover, Chapter 2, Chapter 3 and Chapter 4 are all focused on the use of some novel topological representations of data, while Chapter 6, deals with a separated topic, namely distributional data analysis, sharing with the previous chapters the philosophy of the approach which leads to the novel techinques defined. Chapter 2 and Chapter 3 can be read independently; in this case however, some results exploited in Chapter 3 must be taken for granted since their proofs belong to Chapter 2. Chapter 4 assumes the reader is familiar either with the ideas formulated in Chapter 2 or in Chapter 3. Chapter 6 is a completely standalone chapter.

The separation in chapters, along with their titles, helps the reader in distinguishing between parts devoted to topological data analysis and parts devoted to distributional data analysis. To further help the reader, at the beginning of each chapter we state with a small image the nature of the chapter itself, expressing whether is focused on using/defining topological summaries, or on the developement of tools for non-Euclidean spaces.

The trefoil knot in Figure 1.1(a) indicates a chapter which aims at developing the

topological perspective in the dissertation, while the sketch of a smooth surface, as in Figure 1.1(b), stands for the developement of tools for data analysis outside linear spaces.

**(a)** *Trefoil knot.*



**(b)** *Sketch of a smooth surface.*

**Figure 1.2:** *The two symbols which are used at the beginning of each chapter to guide the reader.*

The trefoil knot is an important object in topology, especially in knot theory, and represents a class, up to homotopy equivalence, of non trivial knots. The surface representation, instead, expresses the more basic operations which one would like to define outside linear spaces: local coordinates to parametrize neighborhoods of points, tangent spaces to obtain a linear structure viable close to the tangent point, and a normal space to talk about covariant derivatives and so geodesics.

Note that these symbols have already been used in the previous sections of the introduction.

## 1.6 Further Comments

The content of the main chapters of the thesis is also part of the following papers:

- Chapter 2: A Metric for Tree-like Topological Summaries (Pegoraro, 2021)
- Chapter 3: Functional Data Representation with Merge Trees (Pegoraro and Secchi, 2021)
- Chapter 4: Frechét Means of Finite Sets of Merge Trees [Preliminary stage]
- Chapter 6: Projected Statistical Methods for Distributional Data on the Real Line with the Wasserstein Metric (Pegoraro and Beraha, 2021)

CHAPTER 2

# A Metric for Tree-Like Topological Summaries

The content of this chapter is also part of the paper Pegoraro (2021).

In this chapter we define a novel metric structure for a family of tree-like topological summaries. This family of objects is a natural combinatoric generalization of merge trees of scalar fields and hierarchical dendrograms. The metric introduced can be computed with a dynamical integer linear programming approach and we showcase its feasibility and the effectiveness of the whole framework with simulated data sets. In particular we stress the versatility of these topological summaries, which prove to be very effective in situation where other topological data analysis tools, like persistence diagrams, can not be meaningfully employed.

## 2.1 Introduction

Topological Data Analysis (TDA) is the name given to an ensemble of techniques which are mainly focused on retrieving topological information from different kinds of data

(Lum et al., 2013). Consider for instance the case of point clouds: the topology of a point cloud itself is quite poor and it would be much more interesting if, using the point cloud, one could gather information about the topological space data was sampled from. Since, in practice, this is often not possible, one can still try to capture the "shape" of the point cloud. The idea of *persistent homology* (PH) (Edelsbrunner and Harer, 2008) is an attempt to do so: using the initial point cloud, a nested sequence of topological spaces is built, which are heavily dependent on the initial point cloud, and PH tracks along this sequence the persistence of the different topological features which appear and disappear. As the name *persistent homology* suggests, the topological features are understood in terms of generators of the homology groups (Hatcher, 2000) taken along the sequence of spaces. One of the fundational results in TDA is that this information can be represented by a set of points on the plane (Edelsbrunner et al., 2002; Zomorodian and Carlsson, 2005), with a point of coordinates $(x, y)$ representing a topological feature being born at time $x$ along the sequence, and disappearing at time $y$. Such representation is called *persistence diagram* (PD). Persistence diagrams can be given a metric structure through the *Bottleneck* and *Wasserstein* metrics, which, despite having good properties in terms of continuity with respect to perturbation of the original data (Cohen-Steiner et al., 2007, 2010), provide badly behaved metric spaces. Various attempts to define tools to work in such spaces have been made (Fasy et al., 2014; Lacombe et al., 2018; Mileyko et al., 2011; Turner et al., 2012), but still it proves to be an hard problem. In order to obtain spaces with better properties and information which is more easily represented in terms of fixed length vectors (needed for many Machine Learning techniques) a number of topological summaries, alternative to PDs, have been proposed, such as: persistence landscapes (Bubenik, 2015), persistence images (Adams et al., 2017) and persistence silhouettes (Chazal et al., 2015).

All the aforementioned machinery has been successfully applied to a great number of problems in a very diverse set of scientific fields: complex shape analysis (MacPherson and Schweinhart, 2010), sensor network coverage (Silva and Ghrist, 2007), protein structures (Gameiro et al., 2014; Kovacev-Nikolic et al., 2016), DNA and RNA structures (Emmett et al., 2015; Rizvi et al., 2017), robotics (Bhattacharya et al., 2015; Pokorny et al., 2015), signal analysis and dynamical systems (Maletić et al., 2015; Perea and Harer, 2013; Perea et al., 2015), materials science (Kramár et al., 2013; Xia et al., 2015), neuroscience (Curto, 2016; Giusti et al., 2016), network analysis (Pal et al., 2017; Sizemore et al., 2015), and even deep learning theory (Hofer et al., 2017; Naitzat et al., 2020).

### Related Works

Close to the definition of persistent homology for $0$ dimensional homology groups, lie the ideas of *merge trees* of functions, *phylogenetic trees* and *hierarchical clustering dendrograms*. Merge trees of functions (Morozov and Weber, 2013) are a particular case of *Reeb Graphs* (Biasotti et al., 2008; Shinagawa et al., 1991), occurring when using

the sublevel sets of a bounded Morse function (Audin et al., 2014) defined on a simply connected domain. Phylogenetic trees and clustering dendrograms are very similar objects which describe the evolution of a set of labels under some similarity measure or agglomerative criterion. Both objects are widely used respectively in phylogenetic and statistics and many complete overviews can be found, for instance see Felsenstein and Felenstein (2004); Garba et al. (2021) for phylogenetic trees and Murtagh and Contreras (2017); Xu and Tian (2015) for clustering dendrograms. Informally speaking, while persistence diagrams record only that, at certain level along a sequence of topological spaces, some path connected components merge, merge trees, phylogenetic trees and clustering dendrograms encode also the information about which components merge with which. Usually tools like phylogenetic trees and clustering dendrograms are used to infer something about a fixed set of labels, for instance an appropriate clustering structure, however, we are more interested in looking at the information they carry as unlabeled objects obtained with different sets of labels. For this reason most of the metrics available for phylogenetic trees and clustering dendrograms are not valuable for our purposes.

In the last years a lot of research sparkled on such topics, starting from the more general case of Reeb graphs, to some more specific works on merge trees. Different but related metrics have been proposed to compare Reeb graphs (Bauer et al., 2014a,b, 2016, 2020; Carrière and Oudot, 2017; De Silva et al., 2016; Di Fabio and Landi, 2012, 2016), which have been shown to posses very interesting properties in terms of Morse functions on manifolds, connecting the combinatorial nature of Reeb Graphs with deformation-invariant characterizations of manifolds which are smooth, compact, orientable and without boundary. On the specific case of merge trees, there has been some research on their computation (Morozov and Weber, 2013; Pascucci and Cole-McLaughlin, 2003) and on using them as visualization tools (Bock et al., 2017; wu and Zhang, 2013), while other works (Beketayev et al., 2014; Morozov et al., 2013) started to build frameworks to analyze sets of merge trees, mainly proposing a suitable metric structure to compare them, as do some recent preprints (Gasparovic et al., 2019; Touli, 2020). The main issue with all the proposed metrics is their computational cost, causing a lack for examples and applications also when approximation algorithms are available (Touli and Wang, 2018). When applications and analysis are carried out (Sridharamurthy et al., 2020), the employed metric does not have suitable properties and thus the authors must resort to a "computational solution to handle instabilities" ((Sridharamurthy et al., 2020), Section 1.2) to use their framework. Along with that, such metrics are difficult to be extended to more general objects than merge trees. Lastly, there is a recent preprint investigating structures lying in between merge trees and persistence diagrams, to avoid computational complexity while retaining some of the additional information provided by such objects (Elkin and Kurlin, 2020).

### Main Contributions

The success of PDs highlighted before, strongly motivates the development of more refined and computable techniques to work with merge trees, phylogenetic trees and clustering dendrograms. Our contribution to such topic is three folded: first we introduce a novel use of tree-like structures as topological summaries with objects called *generalized dendrograms*, second we propose a metric structure for the space of *generalized dendrograms* in the form of a novel edit distance between weighted (in a very broad sense), unlabeled, unordered trees; lastly we develop a dynamical integer programming algorithm to make this metric viable for a good range of applications.

If we consider our framework restricted just to the case of merge trees, the works Bauer et al. (2016); Di Fabio and Landi (2012, 2016) contain a perspective which is very similar to ours, since they define edit distances for Reeb Graphs, and thus, for merge trees. However, at a closer look, the two approaches diverge immediately: the approach in Bauer et al. (2016); Di Fabio and Landi (2012, 2016) is focused of interpreting modifications of the graphs in terms of deformations of the initial topological space, while our is more concerned on the computational advantages offered by accurately defined edit distances and the possibility to extend the metric to more general kinds of trees. Thus, we end up with very different definitions and properties (see Section 2.4 and Remark 6 for more details).

The edit distance we propose starts from usual tree edit distances (Bille, 2005; Tai, 1979) but adds fundamental modifications in order to obtain the properties needed to compare topological information. A simplified but similar definition has already been considered in Koperwas and Walczak (2011), but it is just cited in few lines as a possibility without a real motivation, which lacks any kind of investigation (even whether or not it defines a proper metric). As already stated, instead of modifying other metrics for trees (Billera et al., 2001; Feragen et al., 2012; Wang and Marron, 2007) in order to allow for different sets of leaves with different cardinalities, we follow the path of edit distances because of the computational properties which they often possess, making them suited for dealing with unordered and unlabelled trees (Hong et al., 2017). The computational issues raised by those kind of trees are in fact a primary obstacle to designing feasible algorithms (Hein et al., 1995). Nevertheless, we are able to obtain an Integer Linear Programming (ILP) algorithm which can compute the distance between two binary trees with $N$ and $M$ leaves respectively, by solving $O(N \cdot M)$ ILP problems with $O(N \cdot log(N) \cdot M \cdot log(M))$ variables and $O(N + M)$ constraints.

### Outline

The chapter is organized as follows. In Section 2.2 we describe the main facts that motivate our work. In Section 2.3 we give formal definitions of generalized dendrograms. In high generality, with Section 2.4 we tackle the problem of finding a suitable metric structure to compare those objects. In Section 2.5 we detail how generalized dendrograms can be employed to build new topological summaries. In Section 2.6 we prove

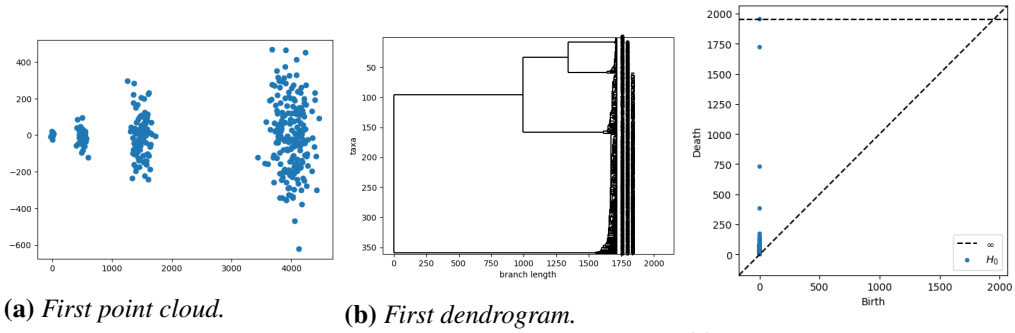
some properties of the metric previously defined, which lead to the algorithm presented in Section 2.7. In Section 2.8 we present some simulations and examples to showcase the effectiveness of the proposed framework and we end up with some conclusions in Section 2.9. Proofs of results are found in Section 2.10.

## 2.2 Main Ideas and Driving Examples

In TDA the main sources of information are sequences of homology groups with field coefficients: using different pipelines a single datum is turned into a sequence of topological spaces, which, in turn, induces - via some homology functor with coefficients in the field $\mathbb{K}$ - a sequence of vector spaces with linear maps which are usually all isomorphisms but for a finite set of points in the sequence. Any such sequence $A_i \xrightarrow{\psi_i^{i+1}} A_{i+1}$ is then turned into a topological summary, for instance a persistence diagram, which completely classifies such objects up to sequence isomorphisms. That is, if for two vector spaces sequences $A_i \xrightarrow{\psi_i^{i+1}} A_{i+1}$ and $B_i \xrightarrow{\eta_i^{i+1}} B_{i+1}$ exists a family of linear isomorphisms $g_i : A_i \to B_i$ such that for all $i$ holds $\eta_i^{i+1} \circ g_i = g_{i+1} \circ \psi_i^{i+1}$, then they are represented by the same persistence diagram. As already highlighted in the introduction, PDs have proven to be useful in a wide variety of tasks. However there might be cases where a more discriminative topological summary is needed, or a summary to which additional information can be meaningfully attached. In Elkin and Kurlin (2020) this topic is discussed and some motivational case studies are carried out, but we want to go further in this direction. To do so we present two simple examples and then try to give some informal intuition of the ideas which are going to be formalized in the following sections.

### 2.2.1 Hierarchical clustering

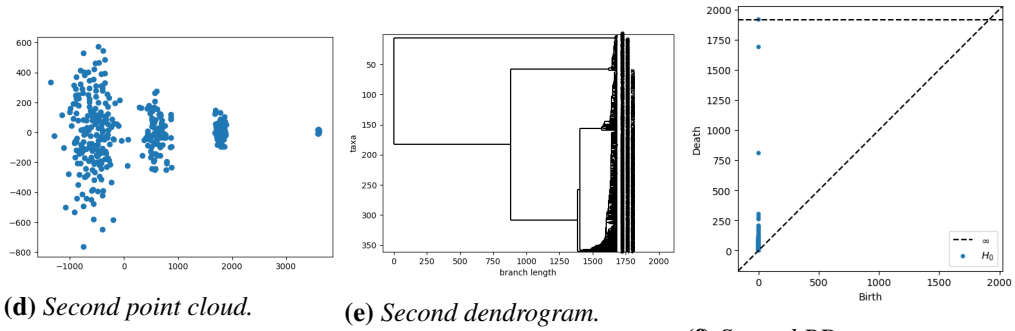
Consider the single linkage dendrograms and the zero dimensional PDs obtained from point clouds as in Figure 2.1 (for a quick introduction to persistence diagrams see Section 2.12). The persistence diagrams (in Figure 2.1(c) and Figure 2.1(f)) are very similar, in fact they simply record that there are four major clusters which merge at similar times across the Vietoris-Rips filtrations (Edelsbrunner and Harer, 2008) of the two point clouds. The hierarchical dendrograms, instead, are clearly very different since they show that in the first case (Figure 2.1(a), Figure 2.1(b), Figure 2.1(c)) the cluster with most points is the one which is more separated from the others in the point cloud; while in the second case (Figure 2.1(d), Figure 2.1(e), Figure 2.1(f)) the two bigger clusters are the first that get merged and the farthest cluster of points on the right could be considered as made by outliers. In many applications it would be important to distinguish between these two scenarios, since the two main clusters get merged at very different heights on the respective dendrograms. We use this example to point out




**(a)** *First point cloud.*

**(b)** *First dendrogram.*

**(c)** *First PD.*

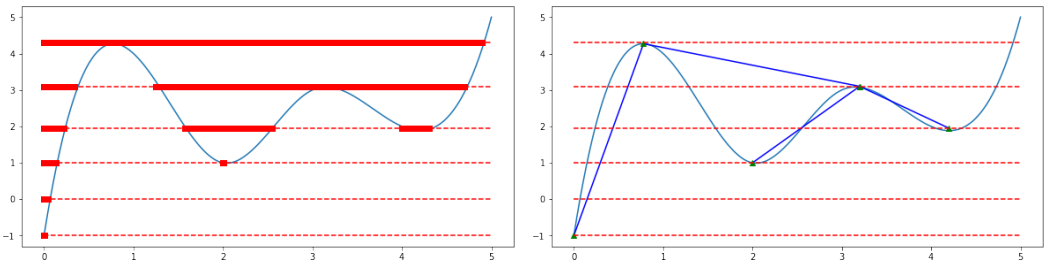



**(d)** *Second point cloud.*

**(e)** *Second dendrogram.*

**(f)** *Second PD.*

**Figure 2.1:** *Data clouds, hierarchical clustering dendrograms and PDs involved in the first example.*



**(a)** *Sublevel sets of a function* **(b)** *A function with its associated merge tree.*

**Figure 2.2:** *Merge Trees of Functions*

another fact: while both dendrograms have as many leaves as there are points in the point clouds, if one attaches to a vertex of the dendrogram the cardinality of the cluster obtained by cutting the edge above the vertex itself, then most of the information contained in the dendrogram could be summarized using a much smaller tree (in terms of number of leaves). For instance one could decide to remove all the vertices associated to clusters whose cardinality is smaller than a certain threshold.

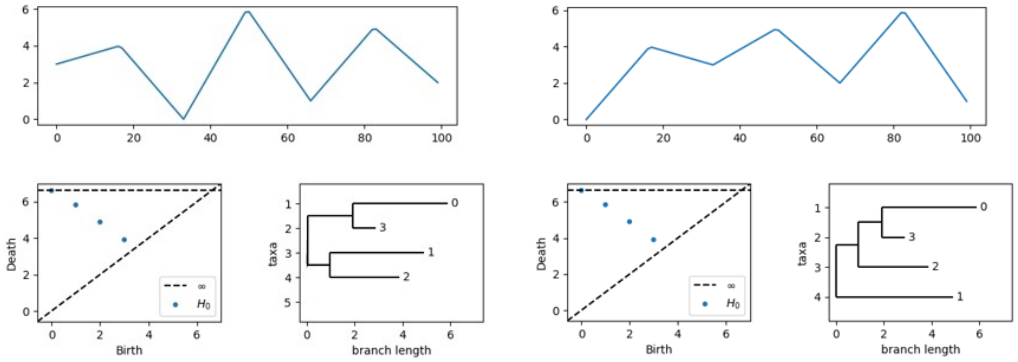
### 2.2.2 Merge Trees of functions

Given a continuous function $f : [a, b] \to \mathbb{R}$ we can extract its merge tree. For a detailed definition of the procedure refer to Morozov and Weber (2013) or Chapter 3. Roughly speaking the merge tree tracks the evolution of the path connected components of the sublevel sets $f^{-1}((-\infty, t])$, for an example see Figure 2.2(b).

Again, we point out two facts. First PDs may not be able to distinguish functions one may wish to distinguish, as made clear by Figure 2.3. Second, Proposition 5 in Chapter 3 states that if one changes the parametrization of a function by means of homeomorphisms, then, both the associated merge tree and persistence diagram do not change. A consequence of such result is that one can linearly shrink or spread the domain of the function $f : [a, b] \to \mathbb{R}$ at will, without changing its merge tree (and PD). There are cases in which such property may be useful but surely there are times when one may want to distinguish if an oscillation lasted for a time interval of $10^{-5}$ or $10^{5}$.

### 2.2.3 Intuitions

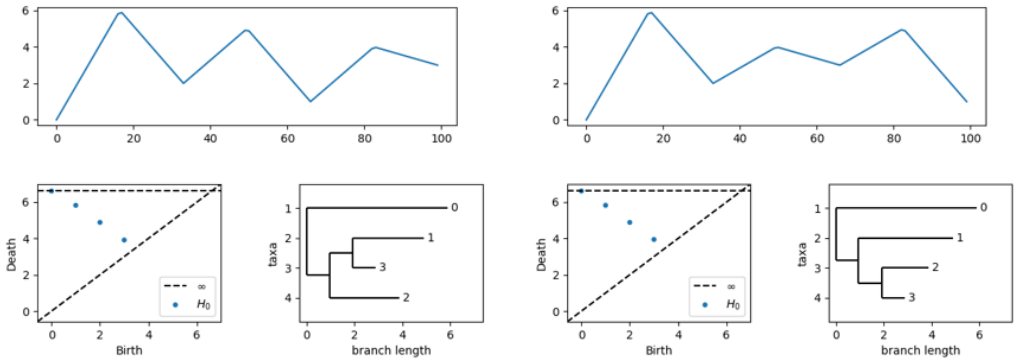
Behind the examples in Section 2.2.1 and Section 2.2.2 there is the following phenomenon. Especially in the case of $0$ dimensional homology, one can naturally fix a basis in the homology vector spaces (for instance the one induced by the connected components) such that the maps in the sequence coherently map the basis fixed in one space into the basis fixed into the following one. Both dendrograms and merge trees capture the information given by such maps and bases along their respective sequences. In addition to that, having fixed a basis inside every homology group, one can gather in-




**(a)** $f_0$




**(b)** $f_1$




**(c)** $f_2$




**(d)** $f_3$

**Figure 2.3:** *We see a comparison of four functions with the same PD in dimension* 0 *but different merge trees. The function is displayed on the first row of each subplot, while on the second we have on the left the PD and on the right the merge tree.*

formation about that basis and follow it at every step along the vector spaces sequence. For instance, in the Vietoris-Rips filtration of a finite point cloud, one could count the number of points in each connected component along the sequence of homology groups.

**Remark 1.** *This idea of gathering additional information along the homology groups, a priori, can be applied also to PDs and the basis which is used to find the diagram representation. But we argue that this cannot be done meaningfully because minimal changes in the sequence of vector spaces, for instance exchanging the birth order of two classes, yields big changes in terms of basis representation, due to the elder rule. For more details see the second example in Section 2.5.3.*

## 2.3 Tree-Like Summaries

In this section we start to formalize our ideas: we start from merge trees, built in a combinatorial fashion, then we define more general objects, called generalized dendrograms, on which we focus for the remaining of the manuscript. To begin with, we state which are the vector spaces sequences we want to work with.

**Definition 1.** *A fixed basis vector spaces filtration is a family of couples* $\{(A_t, a_t)\}_{t\in\mathbb{R}}$ *where* $a_t$ *is a finite dimensional vector space of dimension* $n_t$ *and* $a_t = \{a_1^t, \dots, a_{n_t}^t\}$ *is a basis for* $A_t$*; there are also maps for every* $t < t' \in \mathbb{R}$*:* $\psi_t^{t'} : A_t \to A_{t'}$ *which must satisfy the following conditions:*

1. *given* $t < t' < t''$*, then* $\psi_t^{t'} \circ \psi_{t'}^{t''} = \psi_t^{t''}$*, this is called the cocycle condition;*
2. $\psi_t^{t'}(a_t) \subset a_{t'}$*;*
3. *for any* $t$*,* $\{\psi_t^{t'}\}_{t'}$ *are all isomorphisms but for a finite set of* $t' \in \mathbb{R}$*; such* $t'$ *are called critical points;*
4. *there exists a value* $t^-$ *such that for any* $t < t^-$*,* $(v_t, V_t) = (\varnothing, \{0\})$*;*
5. *there exists a value* $t^+$ *such that, for any* $t^+ \geqslant t$*,* $dim(V_t) = 1$*.*

**Remark 2.** *In Topology and more in general in Category Theory, filtrations (and filtered objects) are obtained with sequences of objects and morphisms (usually monomorphisms). For this reason, with a slight abuse of notation, we use the terms filtration and sequence to refer to ordered sets of objects indexed on the real line, but where "relevant changes" happen only in a finite set of values.*

The the cocycle condition can be exploited to observe some facts about the critical points.

**Remark 3.** *If $t'$ is the first critical point for $t$, that is, it is the smallest value $t'$ bigger than $t$ such that $\psi_t^{t'}$ is not an isomorphism, then for all $t'' \in [t, t')$ the value $t'$ is critical and there are no other critical points for $t''$ in the interval $(t'', t')$. This holds thanks to the cocycle condition. In other words the critical points split the interval $[t^-, t^+]$ in a finite set of intervals $[t_i, t_{i+1})$ where $t_i$ are the critical points. Note that for any $t, t' \in [t_i, t_{i+1})$, $t < t'$, $\psi_t^{t'}$ is an isomorphism.*

Now we define which equivalence classes of sequences we want to work with.

**Definition 2.** *Consider two fixed basis vector spaces filtrations $\mathcal{V} = \{(A_t, a_t)\}_{t\in\mathbb{R}}$ and $\mathcal{W} = \{(B_t, b_t)\}_{t\in\mathbb{R}}$, with maps $\psi_t^{t'} : A_t \to A_{t'}$ and $\eta_t^{t'} : B_t \to B_{t'}$ respectively. A basis preserving isomorphism of sequences $\{g_t\}_{t\in\mathbb{R}}$ is family of linear isomorphisms $g_t : A_t \to B_t$ such that $g_t$ induces a bijection between $a_t$ and $b_t$, and, for all $t$, the following square commutes:*

$$\begin{array}{ccc} A_t & \xrightarrow{\psi_t^{t'}} & A_{t'} \\ \downarrow{\scriptstyle g_t} & & \downarrow{\scriptstyle g_{t'}} \\ B_t & \xrightarrow{\eta_t^{t'}} & B_{t'} \end{array}$$

.

### 2.3.1 Merge Trees

The definition of merge trees is not novel but usually is obtained in a more topological fashion and starting from functions (Morozov and Weber, 2013). Instead we use the more combinatorial approach found in Chapter 3 which we report in the following lines. Such definition relies on graph-based representations of unordered, unlabeled trees, which are called *tree structures* throughout the dissertation.

**Definition 3.** *A tree structure $T$ is given by a set of vertices $V_T$ and set of edges $E_T \subset V_T \times V_T$ which form a connected rooted acyclic graph. The order of a vertex is the number of edges which have that vertex as one of the extremes. Any vertex with an edge connecting it to the root is its child and the root is its father. In this way we recursively define father and children (possibly none) relationships for any vertex on the tree. The vertices with no children are called leaves or taxa. The relationship $father > child$ induces a partial order on $V_T$. Similarly, the edges $E_T$ are given in the form of ordered couples $(a, b)$ with $a < b$. For any vertex $v \in V_T$, $sub_T(v)$ is the subtree of $T$ rooted in $v$, that is the tree structure given by the set of vertices $v' \leqslant v$. If clear from the context we might omit the subscript $T$.*

*A finite tree structure is a tree structure with $V_T$ being a finite set.*

Note that, identifying an edge $(v, v')$ with its lower vertex $v$, gives a bijection between $V_T - \{r_T\}$ and $E_T$, that is $E_T \simeq V_T$ as sets, and thus one can interpret the information associated to vertices as information associated to edges. Given this bijection, we often use $E_T$ to indicate the vertices $v \in V_T - \{r_T\}$, to simplify the notation.

Moreover, we do not want the vertex set of tree structures to carry any relevant structure, since we are going to consistently add pieces of information to a tree structure in a different way. For this reason we treat such objects up to the following isomorphism classes.

**Definition 4.** *Two tree structures $T$ and $T'$ are isomorphic if exists a bijection $g : V_T \to V_{T'}$ that induces a bijection between the edges sets $E_T$ and $E_{T'}$: $(a, b) \mapsto (g(a), g(b))$. Such $g$ is an isomorphism of tree structures.*

Now we can borrow from Chapter 3 the following definition.

**Definition 5.** *A merge tree is a finite tree structure $T$ with a monotone increasing height function $h : V_T \to \mathbb{R}$. Two merge trees $(T, h)$ and $(T', h')$ are isomorphic if $T$ and $T'$ are isomorphic as tree structures and the isomorphism $g : V_T \to V_{T'}$ is such that $h = h' \circ g$. Such $g$ is an isomorphism of merge trees.*

Section 3.2 of Chapter 3 details how, given a function $f : [a, b] \to \mathbb{R}$, with $a, b \in \mathbb{R}$, a merge tree can be used to represent the fixed basis vector spaces filtration given by $A_t = H_0(f^{-1}((-\infty, t]))$ and with $a_t$ being the basis induced by path connected components.

The same procedure can be used to represent any fixed basis vector spaces filtration $\{(A_t, a_t)\}_{t\in\mathbb{R}}$ up to basis preserving isomorphism. The merge tree obtain is unique up to the choice of the vertex set, that is, up to isomorphism of merge trees. The general idea is the following. We consider only the maps $\psi^{t_i}_{t_{i-1}} : A_{t_{i-1}} \to A_{t_i}$, with $\{t_i\}_{i=1,\dots,n}$ being the critical points of the filtration: any time $(\psi^{t_i}_{t_{i-1}})^{-1}(a^{t_i}_s) = \varnothing$, we have a leaf $v$ with $h_T(v) = t_i$, and when we have $\psi^{t_i}_{t_{i-1}}(a^{t_{i-1}}_s) = \psi^{t_i}_{t_{i-1}}(a^{t_{i-1}}_k)$ we have a vertex $v'$ whose children are the vertices of the tree structure associated to the path connected components which merge, with $h_T(v') = t_i$. For more details see Section 2.11. Note that the image of the height function $h_T$ is the set of critical values of the filtration.

### 2.3.2 Generalized Dendrograms

Merge trees are the most natural starting point, since they can be used to represent fixed basis vector spaces filtration up to basis preserving isomorphism. But know we want to take a step forward, collecting and representing other kind of information about those sequences of vector spaces, generalizing merge trees.

Since deciding what kind of information we want to track along a fixed basis vector spaces filtration and deciding how to attach it to the tree-structure may end up overloading the notation and making too restrictive hypotheses, we take a more general approach, developing the theory forgetting about homology groups and fixed bases, but recovering such more specific point of view in Section 2.5.

**Definition 6.** *Given two sets $X$ and $Y$, consider their disjoint union $X \coprod Y$, and a tree structure $T$. A multiplicity function $\varphi$ is a function $\varphi : V_T \to X \coprod Y$, such that $\varphi(E_T) \subset X$ and $\varphi(r_T) \in Y$.*

**Definition 7.** *A generalized dendrogram is a tree structure with a multiplicity function. Two generalized dendrograms $(T, \varphi)$ and $(T', \varphi')$ are isomorphic if there is a bijection $g : V_T \to V_{T'}$ which makes them isomorphic as tree structures and is such that $\varphi(v) = \varphi'(g(v))$.*

## 2.4 Edit Distance for Generalized Dendrograms

The main goal of the following Sections is to propose a computable (pseudo) metric between generalized dendrograms. We want this metric to be suitable to compare topological information, in the sense explained by Section 2.4.2.

### 2.4.1 Edits of Dendrograms

The approach we follow is to define a distance which is inspired by the Tree Edit Distances (Tai, 1979), but with substantial differences in the edit operations. The philosophy of these distances is to allow certain modifications of the base object, called edits, each being associated to a cost, and to define the distance between two objects as the minimal cost that is needed to transform the first object into the second with a finite sequence of edits. Edit distances in fact frequently enjoy some decomposition properties which simplify the calculations (Hong et al., 2017), which are notoriously very heavy (Hein et al., 1995).

First of all, let us make some hypotheses on the multiplicity functions and their codomains.

**Definition 8.** *A set $X$ is called editable if the following conditions are satisfied:*

*(P1)* *$(X, d)$ is a metric space*

*(P2)* *$(X, \oplus, 0)$ is a monoid (that is $X$ has an associative operation $\oplus$ with zero element $0$)*

*(P3)* *the map $d(\cdot, 0) : X \to \mathbb{R}$ is a map of monoids between $(X, \oplus)$ and $(\mathbb{R}, +)$: $d(x \oplus y, 0) = d(0, x) + d(0, y)$.*

*(P4)* *$d$ is $\oplus$ invariant, that is: $d(x, y) = d(z \oplus x, z \oplus y) = d(x \oplus z, y \oplus z)$*

Note that in property (P3), $d(x \oplus y, 0) = d(x, 0) + d(y, 0)$, implies that $x \oplus y \neq 0$. Moreover (P3)-(P4) imply that the points $0$, $x$, $y$ and $x \oplus y$ form a rectangle which can be isometrically embedded in an Euclidean plane with the Manhattan geometry (that is, with the norm $|| \cdot ||_1$): $d(x, x \oplus y) = d(0, y)$, $d(y, x \oplus y) = d(0, x)$ and $d(x \oplus y, 0) = d(0, x) + d(0, y)$.

With these additional pieces of structure there are situations which we want to avoid, because they represent "degenerate" dendrograms which introduce formal complications.

**Definition 9.** *Given an editable space $X$ and a tree-structure $T$, a proper multiplicity function is a multiplicity function $\varphi$ such that $\varphi : E_T \to X$ and $0 \notin \varphi(E_T)$.*

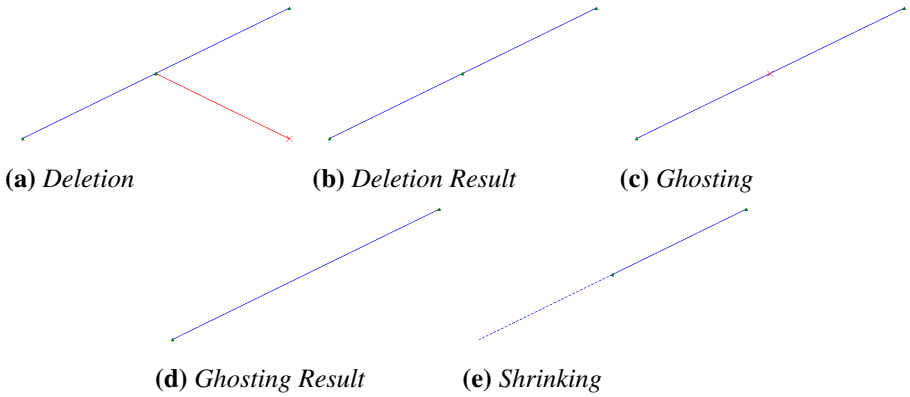
From now on we only work with editable spaces, and we want to consider exclusively proper multiplicity functions. To lighten the notation, however, we omit to write "proper" explicitly.

**Definition 10.** *Given an editable space $X$ and a metric space $Y$, the editable dendrogram space $(\mathcal{D}, Y \coprod X)$ is given by the set of generalized dendrograms with (proper) multiplicity functions with values in $Y \coprod X$.*

Given an editable dendrogram space $(\mathcal{D}, Y \coprod X)$, with $(X, \oplus, 0)$ editable space, we can define our edits.

- We call *shrinking* of a vertex/edge a change of the multiplicity function. The new multiplicity function must be equal to the previous one on all vertices, apart from the "shrunk" one. In other words, for an edge $e$, this means changing the value $\varphi(e)$ with another non zero value in $X$. For the root, this means changing arbitrarily its multiplicity value inside $Y$.
- A *deletion* is an edit with which a vertex/edge is deleted from the dendrogram. Consider an edge $(v_1, v_2)$. The result of deleting $v_1$ is a new tree structure, with the same vertices a part from $v_1$ (the smaller one), and with the father of the deleted vertex which gains all of its children. The inverse of the deletion is the *insertion* of an edge along with its lower vertex. We can insert an edge at a vertex $v$ specifying the name of the new child of $v$, the children of the newly added vertex (that can be either none, or any portion of the children of $v$), and the value of the multiplicity function on the new edge. This edit cannot be done on the root.
- Lastly we define a transformation which eliminates an order two vertex, connecting the two adjacent edges which arrive and depart from it. Suppose we have two edges $e = (v_1, v_2)$ and $e' = (v_2, v_3)$, with $v_1 < v_2 < v_3$. And suppose $v_2$ is of order two. Then, we can remove $v_2$ and merge $e$ and $e'$ into a new edge $e'' = (v_1, v_3)$, with $\varphi(e'') := \varphi(e) \oplus \varphi(e')$. This transformation is called the *ghosting* of the vertex. This edit cannot be done on the root. Its inverse transformation is called the *splitting* of an edge.

A generalized dendrogram $T$ can be edited to obtain another dendrogram, on which one can apply a new edit to obtain a third dendrogram and so on. One can think of this as composing two edits $e_0$, $e_1$ which are not defined on the same dendrogram, since the second edit is defined on the already edited dendrogram. This is what we mean by composition of edits. Any finite composition of edits is called *edit path*. The notations we use are functional notations, even if the edits are not operators, since an edit is not defined on the whole space of dendrograms but on a single dendrogram; for example $e_1 \circ e_0(T)$ means that $T$ is edited with $e_0$, and then $e_0(T)$ with $e_1$.



**(a)** *Deletion* **(b)** *Deletion Result* **(c)** *Ghosting*

**(d)** *Ghosting Result* **(e)** *Shrinking*

**Figure 2.4:** *(a)→(e) form an edit path made by one deletion , one ghosting and a final shrinking, between merge trees.*

### Merge Trees

An example of editable space is $(\mathbb{R}_{\geqslant 0}, +, 0)$ with the metric given by $d(x, y) = |x - y|$. Given a merge tree $T$, with its height function $h_T$, upon replacing $h_T$ with a weight function $w_T$, such that for each edge $e = (v, v') \in E_T$, $w_T(v) = h_T(v') - h_T(v)$, and $w_T(r_T) = h_T(r_T)$, we retrieve a framework to work with merge trees. In fact, by the monotonicity of $h_T$, $w_T$ is a proper multiplicity function. Note that the "map" $h_T \mapsto w_T$ can be naturally inverted, so that from any weight function $w_T : V_T \to \mathbb{R}_{\geqslant 0}$ we can recover a unique height function $h_T$. In Figure 2.4 we can see examples of edit operations for such dendrograms.

### Curves in Editable Spaces

Consider an editable space $X$. Then the space of functions $\{f : \mathbb{R} \to X \big| \int_{\mathbb{R}} d(f(t), 0)dt < \infty\}$ induces an editable space. The monoid operation is defined pointwise: $(f \oplus g)(t) := f(t) \oplus g(t)$ and a pseudo-metric is given by $d(f, g) := \int_{\mathbb{R}} d(f(t), g(t))dt$. If then all functions which differ on zero measure sets (with respect to the Lebesgue measure on $\mathbb{R}$) are identified with an equivalence relationship, this becomes a metric space. The function $d$ is always non negative, so if properties (P3) and (P4) hold pointwise, then they hold also for integrals. For instance we verify (P3) as follows:

$$d(f \oplus g, 0) = \int_{\mathbb{R}} d(f(t) \oplus g(t), 0)dt = \int_{\mathbb{R}} d(f(t), 0) + d(g(t), 0)dt = d(f, 0) + d(g, 0)$$

.

We name such editable space $L_1(\mathbb{R}, X) := \{f : \mathbb{R} \to X \big| \int_{\mathbb{R}} d(f(t), 0)dt < \infty\}/\sim$.

**Finite Products of Spaces**

Consider two editable spaces $X$ and $X'$, that is $(X, \odot, 0_X)$ and $(X', \diamond, 0_{X'})$ satisfying properties (P1)-(P4). Then $(X \times X', \oplus, (0_X, 0_{X'}))$ is an editable space, with $\oplus$ being the component-wise operations $\odot$ and $\diamond$, and the metric $d$ on $X \times X'$ being the sum of the component-wise metrics of $X$ and $X'$.

### 2.4.2 Order 2 vertices

When deleting an edge in a merge tree, the father of the deleted vertex becomes an order two vertex. Such vertex carries no topological information, since the merging that the point was representing, is no more happening (was indeed deleted). This fact gives the intuition that order 2 vertices (a part from the root) are completely irrelevant and must not be taken into account when comparing dendrograms: they appear when nothing interesting happens topologically. Thus, informally speaking, dendrograms "equal" up to order two vertices, should be considered equal. This means that the isomorphism classes considered in Definition 5 and Definition 7 might be "too small" in the sense that one would like to regard as equivalent bigger sets of merge trees or dendrograms. Thanks to the definitions in Section 2.4.1 we can formalize the meaning of "equal up to order 2 vertices".

**Definition 11.** *Generalized dendrograms are equal up to order* 2 *vertices if they become isomorphic after applying a finite number of ghostings or splittings.*

Definition 11 induces an equivalence relationship. The set of generalized dendrograms inside $(\mathcal{D}, X \coprod Y)$ that we want to treat as equal are exactly the equivalence classes given by Definition 11. We call $(\mathcal{D}_2, X \coprod Y)$ the space of equivalence classes of dendrograms in $(\mathcal{D}_2, X \coprod Y)$, equal up to order 2 vertices.

**Definition 12.** *A pseudo-metric on* $(\mathcal{D}, X \coprod Y)$ *which induces a non trivial pseudo-metric on* $(\mathcal{D}_2, X \coprod Y)$ *is called topologically stable.*

In other words a topologically stable pseudo-metric for dendrograms is a (non trivial) pseudo-metric which identifies dendrograms which are equivalent up to order 2 vertices.

### 2.4.3 Edits and Costs

Now we associate to every edit a cost, that is a length measure in the space $(\mathcal{D}, X \coprod Y)$. The costs of the edit operations are defined as follows:

- if, via shrinking, an edge goes from multiplicity $x$ to multiplicity $y$, then the cost of such operation is $d(x, y)$. This holds both for shrinkages happening in $X$ and for shrinkages done on the root, which take place in $Y$;
- for any deletion/insertion of an edge with multiplicity $x$, the cost is equal to $d(x, 0)$;

- the cost of ghosting operations is $|d(x \oplus y, 0) - d(x, 0) - d(y, 0)| = 0$.

**Remark 4.** *With such costs, it would be natural to try to define a family of metrics indexed by integers $p \geqslant 1$ by saying that the costs of compositions are the $p$-th root of sum of the costs of the edit operations to the $p$-th power. But one immediately sees that for any $p > 1$ this has no hope of being a meaningful pseudo metric. In fact consider the case of merge trees (with multiplicity given by the weight function $w_T$) and in particular a tree made by a segment of length $1$. The cost of shrinking it would be $||1||_p = 1$. At the same time one can split it in half with $0$ cost and the cost of shrinking this other tree would be $||(1/2, 1/2)||_p < 1$. Splitting the segment again and again will make its shrinking cost go to $0$. In other words all trees would be in the same equivalence class of the tree with no branches.*

**Definition 13.** *Given two dendrograms $T$ and $T'$ in $(\mathcal{D}, X \coprod Y)$, define:*

- *$\Gamma(T, T')$ as the set of all finite edit paths between $T$ and $T'$;*
- *$cost(\gamma)$ as the sum of the costs of the edits for any $\gamma \in \Gamma(T, T')$;*
- *the dendrogram edit distance as:*

$$d_E(T, T') = inf_{\gamma \in \Gamma(T,T')} cost(\gamma)$$

By definition the triangle inequality and symmetry must hold, but, up to now, this edit distance is intractable; one would have to search for all the possible finite edit paths which connect two dendrograms in order to find the minimal ones. And from Remark 4 we see that is not even obvious that $d_E(T, T') > 0$ for some dendrograms. However, since the cost of ghostings is zero, it is clear that $d_E$ induces a pseudo-metric on classes of dendrograms up to order two vertices.

**Remark 5.** *From the definition of the edit operations and their costs, it is clear that the roots play little to no role in editing a dendrogram: if one wants to turn a dendrogram $T$ in a dendrogram $T'$, he has no choice but shrinking the root $r_T$ to match the multiplicity of $r_{T'}$. So there are no degrees of freedom involved. For this reason, from now on, to lighten the notation, we simply forget $Y$ and the multiplicity of the root and just focus on the weight space $X$. Moreover we always assume to be working in an editable dendrogram space.*

### 2.4.4 Mappings

Now we introduce a fundamental tool, called *mapping*, that, by parametrizing certain sets of edit paths, makes $d_E$ computable and its properties more readily available. The idea of mappings is not novel (Tai, 1979) and often it is a the key ingredient both for proofs and calculations in Tree Edit Distances (Hong et al., 2017). From now on we suppose that in the set of vertices of any dendrogram there are not the letters "D" and

"G" (which are used to indicate "deletion" and "ghosting"). Recall that $E_T$ identifies the vertices $V_T - \{r_T\}$.

A *mapping* between $T$ and $T'$ is a set $M \subset (E_T \cup \{D, G\}) \times (E_{T'} \cup \{D, G\})$ with the following properties:

(M1) consider the projection of the Cartesian product $(E_T \cup \{D, G\}) \times (E_{T'} \cup \{D, G\}) \to (E_T \cup \{D, G\})$; we can restrict this map to $M$ obtaining $\pi_T : M \to (E_T \cup \{D, G\})$. The maps $\pi_T$ and $\pi_{T'}$ are surjective on $E_T \subset (E_T \cup \{D, G\})$ and $E_{T'} \subset (E_{T'} \cup \{D, G\})$ respectively;

(M2) $\pi_T$ and $\pi_{T'}$ are injective;

(M3) $M \cap (V_T \times V_{T'})$ is such that, given $(a, b)$ and $(c, d) \in M \cap (V_T \times V_{T'})$, $a > c$, if and only if $b > d$;

(M4) if $(a, G)$ (or $(G, a)$) is in $M$, let $b_1, .., b_n$ be the children of $a$. Then there is one and only one $i$ such that for all $j \neq i$, for all $x \in V_{sub(b_j)}$, we have $(x, D) \in M$ (respectively $(D, x)$); and there is one and only one $c$ such that $c = max\{x' \in sub(b_i) | (x', y) \in M$ for any $y \in V_{T'}\}$.

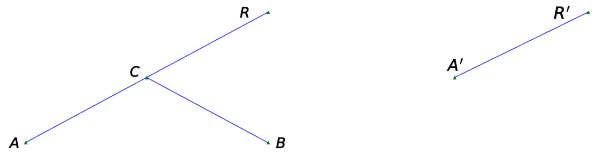
Conditions (M1)-(M2) are asking that every point is assigned to one and only one "transformation"; (M3) ensures that the associations induced by $M \cap (V_T \times V_{T'})$ respect the tree structures of $T$ and $T'$; lastly (M4) means that, once all vertices $v$ appearing in the couples $(v, D)$ or $(D, v)$ in $M$ are deleted, the points which are coupled with $G$ (that is $(a, G)$ or $(G, a)$) are all vertices of order two and therefore they can be ghosted.

Using $M$, we can parametrize a set of edit paths in the dendrogram space, starting from $T$ and ending in $T'$, which are collected under the name $\gamma_M$. The properties of $M$ allow the definition the following edit paths:

- $\gamma_d^T$ is a path made by the deletions to be done in $T$, that is, the couples $(v, D)$, executed in any order. So we obtain $T_d^M = \gamma_d^T(T)$, which, instead, is well defined and not depending on the order of the deletions.
- One then proceeds with ghosting all the vertices $(v, G)$ in $M$, in any order, getting a path $\gamma_g^T$ and the dendrogram $T_M := \gamma_g^T \circ \gamma_d^T(T)$.
- Since all the remaining points in $M$ are couples, the two dendrograms $T'_M$ (defined in the same way as $T_M$, but starting from $T'$) and $T_M$ must be isomorphic as tree structures. This is guaranteed by the properties of $M$. So one can shrink $T_M$ onto $T'_M$, and the composition of the shrinkings, executed in any order is an edit path $\gamma_s^T$.

By definition:

$$\gamma_s^T \circ \gamma_g^T \circ \gamma_d^T(T) = T'_M,$$




**(a)** $T$ **(b)** $T'$

**Figure 2.5:** *The edit path in Figure 2.4 is represented by the mapping* $M = \{(B, D), (C, G), (A, A'), (R, R')\}$ *between* $T$ *and* $T'$*. Figure 2.4(b) represents* $T_d^M$*, Figure 2.4(d) presents* $T_M$*; in this case* $T'_M = T'$*.*

and:

$$(\gamma_d^{T'})^{-1} \circ (\gamma_g^{T'})^{-1} \circ \gamma_s^T \circ \gamma_g^T \circ \gamma_d^T(T) = T'$$

.

where the inverse of an edit path is thought as the composition of the inverses of the single edit operations, taken in the inverse order.

Lastly, we call $\gamma_M$ the set of all possible edit paths:

$$(\gamma_d^{T'})^{-1} \circ (\gamma_g^{T'})^{-1} \circ \gamma_s^T \circ \gamma_g^T \circ \gamma_d^T.$$

obtained by changing the order in which the edit operations are executed inside $\gamma_d$, $\gamma_g$ and $\gamma_s$. Observe that, even if $\gamma_M$ is a set of paths, its cost is well defined:

$$cost(M) := cost(\gamma_M) = cost(\gamma_d^T) + cost(\gamma_s^T) + cost(\gamma_d^{T'}).$$

See Figure 2.5 for an example of a mapping between merge trees.

Before moving on, we fix some notation and call $Mapp(T, T')$ the set of all mappings between $T$ and $T'$. This set is never empty, in fact $M = \{(v, D) : v \in E_T\} \cup \{(D, v') : v' \in E_{T'}\}$ is always a mapping between $T$ and $T'$. In other words one can always delete all the edges of a generalized dendrogram, and then insert all the edges of the other.

**Theorem 1** (Main Theorem)**.** *Given two generalized dendrograms* $T$ *and* $T'$*, for every finite edit path* $\gamma$*, exists a mapping* $M \in Mapp(T, T')$ *such that* $cost(M) \leqslant cost(\gamma)$*.*

A first Corollary immediately follows.

**Corollary 1.** *Since* $Mapp(T, T')$ *is a finite set we have the following well defined pseudo-metric:*

$$d_E(T, T') = inf\{cost(\gamma) | \gamma \in \Gamma(T, T')\} = min\{cost(M) | M \in Mapp(T, T')\}$$

*which we will refer to as the edit distance between* $T$ *and* $T'$*.*

A second Corollary is obtained observing that, if a mapping has cost equal to zero, then it must contain only ghostings.

**Corollary 2.** *Given $T$ and $T'$ dendrograms, $d_E(T, T') = 0$ if and only if $T$ and $T'$ are equal up to order $2$ vertices. In other words $d_E$ is a metric for generalized dendrograms considered up to order $2$ vertices.*

**Remark 6.** *If we compare the definitions and the results carried out in this section, with the ones in Bauer et al. (2016); Di Fabio and Landi (2012, 2016), we can recognize the different perspectives with which the different edit distances have been developed: in the cited works, the authors are more focused on transformations of the base topological space, while we are focused on local modifications of dendrograms. In fact, the available edit operations are different: in Bauer et al. (2016); Di Fabio and Landi (2012, 2016) there are six kinds of edits to be done on edges, along with their inverses, which avoid having to deal with the removal of single vertices, situation which, instead, we treat with ghostings. Moreover, even in the case of merge trees, there are some edits in Bauer et al. (2016); Di Fabio and Landi (2012, 2016) which cannot be seen as modifications involving just a single vertex, and this makes difficult to employ something like the mappings as we define, since a mapping is solely based on the fact that we can completely encode any edit with a couple of elements: being it two vertices (of different dendrograms) or a vertex and a letter (either "D" or "G").*

*We can appreciate the different behaviours of the metrics looking at the* stability *results with respect to uniform convergence of functions: comparing for instance Theorem 28 in Di Fabio and Landi (2016) and Theorem 1 in Chapter 3 we see that the behaviour of the two metrics is very different, with the metric in Bauer et al. (2016); Di Fabio and Landi (2012, 2016) being more stable with respect to* sup *norm between functions.*

## 2.5 Back to Vector Spaces Filtrations

At this point we go back to fixed basis vector spaces filtrations and we employ the machinery defined in Section 2.3.2 and Section 2.4 to extract information from such families of vector spaces. We want to define a pipeline to build proper multiplicity functions with values in an editable space, obtaining dendrograms which in some sense generalize merge trees. The precise meaning of "generalizing" merge trees is the following: starting from two fixed basis vector spaces filtrations, if we obtain generalized dendrograms which are isomorphic, we ask that also their merge trees are isomorphic.

Consider $\{(A_t, a_t)\}_{t\in\mathbb{R}}$ fixed basis vector spaces filtration with maps $\psi_t^{t'} : A_t \to A_{t'}$. An *information function* for $\{(A_t, a_t)\}_{t\in\mathbb{R}}$ is a function $\Theta : \mathcal{B} \to X$ such that $a_t \subset \mathcal{B}$ for all $t \in \mathbb{R}$, with $X$ being an editable space. One should think at $\Theta$ as a kind of "sufficient statistic" to be extracted from the fixed basis $a_t$: it is the information one wants to extract from the elements of the basis at time $t$ along the chosen filtration and it must be carefully designed depending on the aim of the analysis.

Consider the merge tree obtained from $\{(A_t, a_t)\}_{t\in\mathbb{R}}$ with its tree structure $T$ and its height function $h_T$. Moreover consider an edge $e = (v, v')$, with $t_i = h_T(v)$ and

$t_j = h_T(v')$. We know by construction that there is a basis element, which we call $a_e^{t_i} \in a_{t_i}$, associated to $v$, which is such that $(\psi_{t_i}^{t'})^{-1}(\psi_{t_i}^{t'}(a_e^{t_i})) = \{a_e^{t_i}\}$ for all $t' \in [t_i, t_j)$. We define the multiplicity function $\varphi_T^\Theta$ so that $\varphi_T^\Theta(e) : \mathbb{R} \to X$ is defined as follows:

$$\varphi_T^\Theta(e)(t') = \Theta(\psi_{t_i}^{t'}(a_e^{t_i}))$$

for all $t' \in [t_i, t_j)$, and $\varphi_T^\Theta(e)(t') = 0$ otherwise.

**Definition 14.** *Given $\mathcal{S}$ a set of fixed basis vector spaces filtrations and $X$ editable space, an $\mathcal{S}$-proper information function $\Theta : \mathcal{B} \to X$, with $X$ editable space, is a function such that:*

- *for every $\{(A_t, a_t)\}_{t\in\mathbb{R}} \in \mathcal{S}$, $a_t \subset \mathcal{B}$*
- *$\varphi_T^\Theta$ is a proper multiplicity function with values in $L_1(\mathbb{R}, X)$ for every $T \in \mathcal{S}$*
- *if $(T, \varphi_T^\Theta)$ and $(T', \varphi_{T'}^\Theta)$ obtained from two elements of $\mathcal{S}$ are isomorphic as generalized dendrograms, then the merge trees $(T, h_T)$ and $(T', h_{T'})$ associated to the same filtrations are isomorphic as merge trees.*

Since $\varphi_T^\Theta(e)$ is by construction zero outside a compact interval, there are many natural conditions to be required for $\Theta$ so that $\varphi_T^\Theta$ is a multiplicity function with values in the editable space $L_1(\mathbb{R}, X)$. For instance we could ask that $d(\Theta(\cdot), 0)$ is bounded by some positive constant. Similarly, if we want $\varphi_T^\Theta$ to be a proper multiplicity function, it is enough that $\Theta(\psi_{t_i}^{t'}(a_e^{t_i})) = 0$ only for $t'$ belonging to measure zero subsets (wrt Lebesgue measure) of $[t_i, t_j]$. Both conditions, as well as the last one requested by Definition 14, which, again, has to do with the zeros of the function $\Theta$, can be attained without much effort in many interesting situations, as shown in the upcoming examples.

Note that the operation of ghosting a vertex, with this structure, assumes a quite natural form. Suppose we have two edges $e = (v_1, v_2)$ and $e' = (v_2, v_3)$, with $v_1 < v_2 < v_3$ and $v_2$ of order two. We have $\varphi_T^\Theta(e)$ with support on $[t_{v_1}, t_{v_2}]$ and $\varphi_T^\Theta(e')$ with support on $[t_{v_2}, t_{v_3}]$, with $t_{v_i}$ being $h_T(v_i)$. If we ghost $v_2$ obtaining $e'' = (v_1, v_3)$, then $\varphi(e'') = \varphi_T^\Theta(e) + \varphi_T^\Theta(e')$ is supported on $[t_{v_1}, t_{v_3}]$ and is such that $\varphi_T^\Theta(e'')(t) = \varphi_T^\Theta(e)(t)$ on $[t_{v_1}, t_{v_2}]$ and $\varphi_T^\Theta(e'')(t) = \varphi_T^\Theta(e')(t)$ on $[t_{v_2}, t_{v_3}]$. Which means that we track down the information collected by $\Theta$ as if $v_2$ did not exist.

**Remark 7.** *From Definition 1, it is clear that the values $t^-$ and $t^+$ are not unique for a fixed basis vector spaces filtration. When building merge trees this is not a concern, because all relevant topological changes happen between the minimum and the maximum critical values. When tracking down some kind of information with $\Theta$, it may happen, for instance, that it is valuable to collect values of $\Theta(a_i^t)$ for $t > max_{i=1,\dots,n} t_i$, and thus one can fix $t^-$ and $t^+$ according to the aim of the analysis. In particular $t^- = min_{i=1,\dots,n} t_i$ is a sensible choice in most cases, since if $t^- < min_{i=1,\dots,n} t_i$, then $a_i^{t^-} = \varnothing$. Instead, fixing $t^+ > max_{i=1,\dots,n} t_i$ can be useful for instance when considering sublevel set filtrations of functions, as in Section 2.5.3. In this case, one should consider $t^+$ as being another critical value for the filtration, so that information is tracked with $\Theta$ up to $t^+$.*

**Remark 8.** *The one presented in this Section is not the only way to design multiplicity functions which track some kind of information along a fixed basis vector spaces filtration, but for the aforementioned reasons and, as motivated by the following examples, it is a quite natural and flexible framework to consider.*

### 2.5.1 Merge Trees

Consider the special case the constant function $\Theta : Sets \to \mathbb{R}_{\geqslant 0}$, such that $\Theta(s) = 1$ for all sets $s$, that is $\varphi_T^{\Theta}(e) = \chi_{[t_i,t_j)}$ for an edge $e$ spanning from height $t_i$ to height $t_j$, with $\chi_I$ being the characteristic function over the set $I \subset \mathbb{R}$. If two dendrograms $(T, \varphi_T^{\Theta})$ and $(T', \varphi_{T'}^{\Theta})$ obtained with such information function are isomorphic, then they must have isomorphic tree structures and the associated fixed basis vector spaces filtrations must share all critical values. Otherwise, for any isomorphism of dendrograms $\eta : E_T \to E_{T'}$, the functions $\varphi_T^{\Theta}(e)$ and $\varphi_{T'}^{\Theta}(e')$, for at least one couple of edges such that $\eta(e) = e'$, are characteristic functions over different intervals of the form $[a, b)$ and thus their distance in $L_1(\mathbb{R}, \mathbb{R}_{\geqslant 0})$ is positive. This immediately implies that $\Theta = 1$ is a proper information function with values in $\mathbb{R}_{\geqslant 0}$ for all fixed basis vector spaces filtrations. With such function one can recover the information contained in merge trees: isomorphism between dendrograms implies isomorphism of merge trees and viceversa. These bijections between merge trees and dendrograms induced by $\Theta = 1$, however, are not isometries wrt the distance $d_E$.

**Example** Consider for instance the functions $f = ||x| - 1|$ and $g = ||x| - 1| + 1$, both defined on the interval $[-2, 2]$. See Figure 2.6(a). Let $A_t = H_0(f^{-1}((-\infty, t]))$ and $B_t = H_0(g^{-1}((-\infty, t]))$, both with the bases given by path connected components.

For the sequence $\{(A_t, a_t)\}_{t\in\mathbb{R}}$ we can fix $t^- = 0$ and $t^+ = 1$. For any $t \in [t^-, t^+]$, $a_t = \{a_1^t, a_{-1}^t\}$, with $a_1^t = [1-t, 1+t]$ and $a_{-1}^t = [-1-t, -1+t]$. The critical points are $t_0 = t^-$ and $t_1 = t^+$. Thus the merge tree $(T, h_T)$ associated to $\{(A_t, a_t)\}_{t\in\mathbb{R}}$ has a tree structure given by a root with two children being the leaves. We represent this with the vertex set $\{v_1, v_{-1}, r_T\}$ and edges $e_1 = (v_1, r_T)$ and $e_2 = (v_{-1}, r_T)$. The height function has values $h_T(v_1) = h_T(v_{-1}) = t^- = 0$ and $h_T(r_T) = t^+ = 1$. See Figure 2.6(b). The multiplicity function $\varphi_T^{\Theta}$, instead, is defined as follows: $\varphi_T^{\Theta}(e_1) = \varphi_T^{\Theta}(e_2) = \chi_{[0,1)}$ and $\varphi_T^{\Theta}(r_T) = \chi_{\{1\}}$.

In a similar fashion the sequence $\{(B_t, b_t)\}_{t\in\mathbb{R}}$ has $t^- = 1$ and $t^+ = 2$. For any $t \in [t^-, t^+]$, $b_t = \{b_1^t, b_{-1}^t\}$, with $b_1^t = [1 - t, 1 + t]$ and $b_{-1}^t = [-1 - t, -1 + t]$. The critical points are $t_0 = t^-$ and $t_1 = t^+$. Again the merge tree $(T', h_{T'})$ associated to $\{(B_t, b_t)\}_{t\in\mathbb{R}}$ has a tree structure given by a root with two children being the leaves. We fix the vertex set $\{w_1, w_{-1}, r_{T'}\}$ and the edges $e'_1 = (w_1, r_{T'})$ and $e'_2 = (w_{-1}, r_{T'})$. The height function has values $h_{T'}(w_1) = h_{T'}(w_{-1}) = t^- = 1$ and $h_{T'}(r_{T'}) = t^+ = 2$. The multiplicity function $\varphi_{T'}^{\Theta}$, instead, is defined as follows: $\varphi_{T'}^{\Theta}(e'_1) = \varphi_{T'}^{\Theta}(e'_2) = \chi_{[1,2)}$ and $\varphi_{T'}^{\Theta}(r_{T'}) = \chi_{\{2\}}$.

Consider the mapping $M = \{(v_1, w_1), (v_{-1}, w_{-1}), (r_T, r_{T'})\}$. Its easy to check, be-

cause of the small size of the sets $V_T$ and $V_{T'}$, that this is a minimizing mapping both for the merge trees $(T, w_T)$ and $(T', w_{T'})$, and for the dendrograms $(T, \varphi_T^\Theta)$ and $(T', \varphi_{T'}^\Theta)$.

The cost of $M$ for the merge trees is:

$$\sum_{i=-1,1} |w_T(v_i) - w_{T'}(w_i)| + |w_T(r_T) - w_{T'}(r_{T'})| = 0 + 0 + 1$$

while for the dendrograms is

$$\sum_{i=-1,1} ||\varphi_T^\Theta(v_i) - \varphi_T^\Theta(w_i)||_1 + ||\varphi_T^\Theta(r_T) - \varphi_T^\Theta(r_{T'})||_1 = ||\chi_{[0,2)}||_1 + ||\chi_{[0,2)}||_1 + 0 = 4$$

### 2.5.2 Clustering Dendrograms

Consider now the case of an agglomerative clustering dendrogram $(T, h_T)$ built on a finite set $\{x_1, \ldots, x_n\}$ with some linkage rule. We can look at clustering dendrograms as the merge trees associated to the filtration given by $t^- = 0$ and for $t \geqslant 0$, $A_t$ is the vector space generated by the clusters obtained by cutting the dendrogram at height $t$. The basis is the one given by $a_t = \{\{x_{1,1}, \ldots, x_{1,n_1}\}, \ldots, \{x_{k,1}, \ldots, x_{k,n_k}\}\}$ where $\{x_{j,1}, \ldots, x_{j,n_j}\}$ is the $j$-th cluster, which has cardinality $n_j$, obtained by cutting the dendrogram at height $t$. We call *clustering filtrations* all the fixed basis vector spaces filtrations obtained from agglomerative clustering dendrograms following the procedure just outlined.

A sensible information that one may want to track down along $\{(A_t, a_t)\}_{t\in\mathbb{R}}$ is the cardinality of the clusters. Thus we can take $\Theta : FSets \to \mathbb{R}_{\geqslant 0}$, defined on all finite sets ($Fsets$), such that $\Theta(\{x_{j,1}, \ldots, x_{j,n_j}\}) = n_j$. Clearly, for a clustering filtration on $n$ elements, $1 \leqslant \Theta \leqslant n$ and so $\varphi_T^\Theta(e) \in L_1(\mathbb{R}, \mathbb{R}_{\geqslant}0)$. Note that $\varphi_T^\Theta(e) = m\chi_{[t_i,t_j)}$, for some positive cardinality $m$ and some critical points $t_i, t_j$. Thus, $\Theta(\{x_{j,1}, \ldots, x_{j,n_j}\}) = n_j$ is a proper family of information functions for all clustering filtrations.

**Example** Consider the finite set $\{v_{-1} = -1, v_0 = 0, v_2 = 2\}$ and build the single linkage hierarchical dendrogram of such set using the euclidean metric. The filtration obtained from such hierarchical dendrogram is $A_t \simeq \mathbb{K}^3$ for $t \in [0, 1)$, $A_t \simeq \mathbb{K}^2$ for $t \in [1, 2)$ and $A_t \simeq \mathbb{K}$ for $t \geqslant t^+ = 2$. The fixed bases are $a_t = \{\{v_{-1}\}, \{v_0\}, \{v_2\}\}$ for $t \in [0, 1)$, $a_t = \{\{v_{-1}, v_0\}, \{v_2\}\}$ for $t \in [1, 2)$ and $a_t = \{\{v_{-1}, v_0, v_2\}\}$ for $t \geqslant t^+ = 2$. The associated merge tree $(T, h_T)$ - see Figure 2.6(c) - can be represented with the vertex set $V_T = \{\{v_{-1}\}, \{v_0\}, \{v_2\}, \{v_{-1}, v_0\}, r_T\}$. The leaves are $\{v_{-1}\}, \{v_0\}$ and $\{v_2\}$; the children of $\{v_{-1}, v_0\}$ are $\{v_{-1}\}$ and $\{v_0\}$, and the ones of $r_T$ are $\{v_{-1}, v_0\}$ and $\{v_2\}$. The height function $h_T$ is given by $h_T(\{v_i\}) = 0$ for $i = -1, 0, 2$, $h_T(\{v_{-1}, v_0\}) = 1$ and $h_T(r_T) = 2$. The multiplicity function $\varphi_T^\Theta$ is thus the following: $\varphi_T^\Theta(\{v_i\}) = \chi_{[0,1)}$ for $i = -1, 0$, $\varphi_T^\Theta(\{v_2\}) = \chi_{[0,2)}$, $\varphi_T^\Theta(\{v_{-1}, v_0\}) = 2\chi_{[1,2)}$ and $\varphi_T^\Theta(r_T) = 3\chi_{\{2\}}$. See Figure 2.6(d).

### 2.5.3 Dendrograms of Functions

Now turn to the situation showcased in Section 2.2.2. Consider $U \subset \mathbb{R}^m$ convex bounded open set, with $\overline{U}$ being its topological closure, and let $\mathcal{L}$ be the Lebesgue measure in $\mathbb{R}^m$. Let $f : \overline{U} \to \mathbb{R}$ be a *tame* (Chazal et al., 2016) continuous function. Consider the sublevel set filtration $A_t = H_0(f^{-1}((-\infty, t]))$ with $a_t = \{U_1^t, \dots, U_n^t\}$ being the path connected components of $f^{-1}((-\infty, t])$. Here the tameness condition is simply asking that $a_t$ is a finite set for every $t$. Call $\psi_t^{t'}$ the functions $\psi_t^{t'} : A_t \to A_{t'}$. We choose as information function $\Theta = \mathcal{L}$, that is: $\Theta(U_i^t) = \mathcal{L}(U_i^t)$. We can set $t^- = \inf_U f$ and $t^+ = \sup_U f$. Let $(T, h_T)$ being the merge tree representing $\{(A_t, a_t)\}_{t\in\mathbb{R}}$, and $\varphi_T^\Theta$ the associated multiplicity function. Being $f$ continuous, for and edge $e = (v, v') \in E_T$ spanning from height $h_T(v) = t_i$ to $h_T(v') = t_j$, now we prove that $\varphi_T^\Theta(e) > 0$ on $(t_i, t_j)$. We now that $v$ is associated to a connected component $U_k^{t_i}$, for some $k$. If $v$ represents the merging of two or more path connected components $U_{k_1}^{t_i-\varepsilon}$ and $U_{k_2}^{t_i-\varepsilon}$, for some small $\varepsilon > 0$, with $\mathcal{L}(U_{k_1}^{t_i-\varepsilon}), \mathcal{L}(U_{k_2}^{t_i-\varepsilon}) > 0$, then, since $U_{k_1}^{t_i-\varepsilon}, U_{k_2}^{t_i-\varepsilon} \subset U_k^{t_i}$, we have $\mathcal{L}(U_k^{t_i}) > 0$. Thus if we prove the statement for $v$ leaf, we are done.

So, suppose $v$ is a leaf and consider $x_0 \in U_k^{t_i}$. We know $f(x_0) = t_i$. By the continuity of $f$, for every $\varepsilon > 0$ there is $\delta > 0$ such that if $||x - x_0|| < \delta$, then $f(x_0) \leqslant f(x) < f(x_0) + \varepsilon$. Since $\{x \in \overline{U} \big| ||x - x_0|| < \delta\}$ is convex (and so path connected), then it is contained in $\psi_{t_i}^{t_i+\varepsilon}(U_k^{t_i})$. Moreover, since it contains the non-empty open set $\{x \in U \big| ||x - x_0|| < \delta\}$, we have $\mathcal{L}(\psi_{t_i}^{t_i+\varepsilon}(U_k^{t_i})) > 0$ for every $\varepsilon > 0$. As a consequence, $\text{supp}(\varphi_T^\Theta(e)) = [t_i, t_j]$. Putting the pieces together this means that $\Theta = \mathcal{L}$ is a proper information function for sublevel set filtrations obtained from real valued, bounded, tame functions defined over the closure of convex, bounded, open subsets of $\mathbb{R}^n$.

**Example** Consider again the function $f = ||x| - 1|$ defined on the interval $[-2, 2]$. Let $A_t = H_0(f^{-1}((-\infty, t]))$ with the bases given by path connected components. The Example in Section 2.5.1 shows how to obtain the merge tree $(T, h_T)$ associated to the sequence $\{(A_t, a_t)\}_{t\in\mathbb{R}}$. Using the same notation of Section 2.5.1, now we obtain the multiplicity functions $\varphi_T^\Theta(e_i)$, with $\Theta$ being the Lebesgue measure as just discussed.

We then have $\varphi_T^\Theta(e_1) = |1 + t - 1 + t| = 2t$ for $t \in [0, 1)$, and 0 otherwise. Clearly $\varphi_T^\Theta(e_1) = \varphi_T^\Theta(e_2)$. Lastly $\varphi_T^\Theta(r_T) = 4\chi_{\{2\}}$.

**Example** Lastly we consider the following functions defined on $[-1, 2]$: $f(x) = |x - 1| + \varepsilon$ if $x \geqslant 0$ and $f(x) = |2x - 1|$ if $x < 0$; while $g(x) = |x - 1|$ if $x \geqslant 0$ and $g(x) = |2x - 1| + \varepsilon$ if $x < 0$ for a fixed $\varepsilon > 0$; as in Figure 2.6(e). Let $(T, h_T)$ and $(T', h_{T'})$ be the merge trees associated to the sublevel set filtrations of $f$ and $g$; moreover let $\varphi_T^\Theta$ and $\varphi_{T'}^\Theta$ the two respective multiplicity functions with $\Theta$ being the Lebesgue measure on $\mathbb{R}$. Note that $||f - g|| = 2\varepsilon$. The local minima of the functions are the points $\{-0.5, 1\}$, with $f(-0.5) = 0$, $f(1) = \varepsilon$, $g(-0.5) = \varepsilon$ and $g(1) = 0$. Thus the merge trees have isomorphic tree structures: we represent $T$ with the vertex set

$\{v_{-0.5}, v_1, r_T\}$ and edges $\{(v_{-0.5}, r_T), (v_1, r_T)\}$; and $T'$ with vertices $\{v_{-0.5}, v_1, r_{T'}\}$ and edges $\{(v_{-0.5}, r_{T'}), (v_1, r_{T'})\}$. The height functions are the following: $h_T(v_{-0.5}) = 0$, $h_{T'}(v_{-0.5}) = \varepsilon$, $h_T(v_1) = \varepsilon$, $h_{T'}(v_1) = 0$ and $h_T(r_T) = h_{T'}(r_{T'}) = 1 + \varepsilon$.

Lastly, the multiplicity functions (see Figure 2.6(f)) are given by: $\varphi^{\Theta}_T(v_{-0.5})(t) = t\chi_{[0,1)} + \chi_{[1,1+\varepsilon)}$, $\varphi^{\Theta}_T(v_1)(t) = 2(t-\varepsilon)\chi_{[\varepsilon,1+\varepsilon)}$ and $\varphi^{\Theta}_{T'}(v_{-0.5})(t) = (t-\varepsilon)\chi_{[\varepsilon,1+\varepsilon)}$ and $\varphi^{\Theta}_{T'}(v_1)(t) = 2t \cdot \chi_{[0,1)} + 2\chi_{[1,1+\varepsilon)}$.

The zero-dimensional persistence diagram associated to $f$ (we name it $PD_0(f)$) is given by a point with coordinates $(0, +\infty)$, associated to the connected component $[-t/2 - 0.5, t/2 - 0.5]$ which is born at $t = 0$, and the point $(\varepsilon, 1 + \varepsilon)$, associated to the component $[1 - (t - \varepsilon), 1 + (t - \varepsilon)]$, born at level $t = \varepsilon$ and "dying" at level $t = 1 + \varepsilon$, due to the elder rule, since it merges an older component, being the other component born at a lower level.

For the function $g$, the persistence diagram $PD_0(g)$ is made by the same points, but the situation is in some sense "reversed". In fact, the point $(0, +\infty)$ is associated to the connected component "centered" in 1, which is $[1 - t, 1 + t]$, and the point $(\varepsilon, 1 + \varepsilon)$, is associated to the component "centered" in $0.5$, that is $[-(t-\varepsilon)/2 - 0.5, (t+\varepsilon)/2 - 0.5]$.

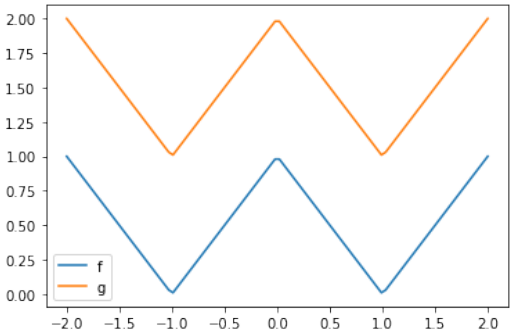
The consequence of this change in the associations between points and the components originating the points of the diagrams is that the information regarding the two components, end up being associated to very different spatial locations in the two diagrams: $(0, +\infty)$ and $(\varepsilon, 1 + \varepsilon)$. And this holds for every $\varepsilon > 0$. Thus it seems very hard to design a way to "enrich" $PD_0(f)$ and $PD_0(g)$ with additional information, originating the "enriched diagrams" $D_f$ and $D_g$, respectively, and design a suitable metric $d$, so that $d(D_f, D_g) \to 0$ as $\varepsilon \to 0$.

Instead, if we consider the mapping $M = \{(v_{-0.5}, v_{-0.5}), (v_1, v_1), (r_T, r_{T'})\}$ we have $d_E((T, \varphi^{\Theta}_T), (T', \varphi^{\Theta}_{T'})) \leqslant cost(M) = 3\varepsilon$. Thus it is very likely that some kinds of continuity/"stability" results, depending on the application, can be proven with our framework, while it seems much harder to do the same for persistence diagrams.

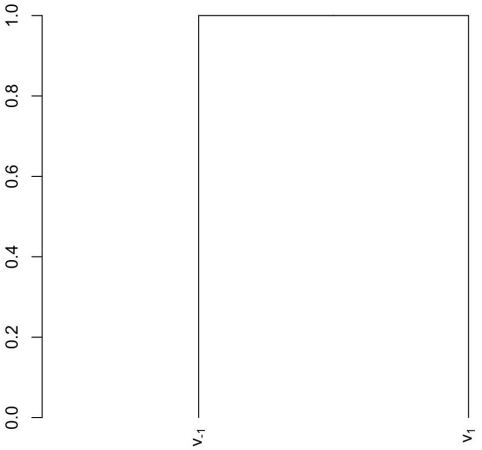
**Remark 9.** *In the previous Sections we presented three frameworks dealing with merge trees, clustering dendrograms and sublevel sets filtrations of functions. More general and complex frameworks can be defined, for instance we could consider suitable functions defined on Riemannian manifolds, with $\Theta$ being the Riemannian volume. Similarly, instead of taking information functions with values in $X = \mathbb{R}_{\geqslant 0}$, we could design functions with values in other editable spaces, such as $\mathbb{R}_{\geqslant 0} \times \mathbb{R}_{\geqslant 0}$.*

## 2.6 Decomposition Properties and Optimization Problems

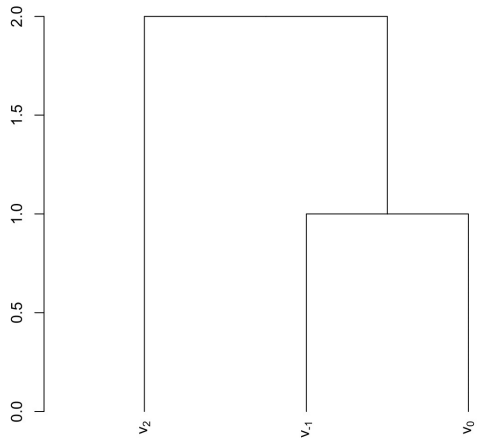
In this section we develop some results and formulations needed to obtain the algorithm presented in Section 2.7. In Section 2.6.1 we prove the theoretical result that allows to recursively split up the calculations (following ideas found in Hong et al. (2017)) and in Section 2.6.2 such result is translated in terms of integer optimization problems.



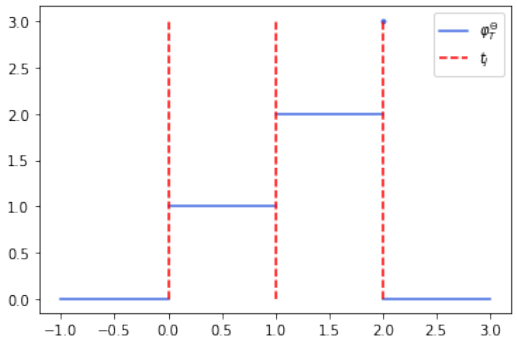

**(a)** *The functions $f$ and $g$ in the Example in Section 2.5.1.*



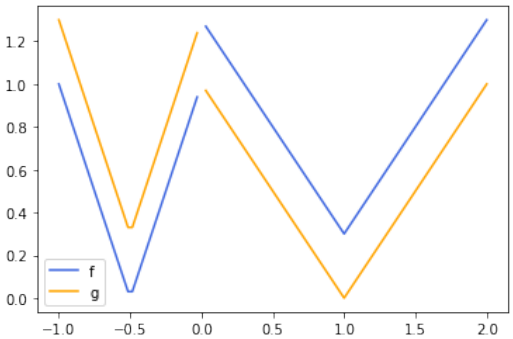

**(b)** *A representation of $(T, h_T)$, see the Example in Section 2.5.1.*



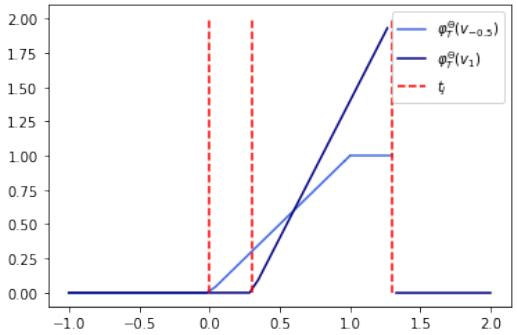

**(c)** *Single linkage clustering dendrogram referring to the Example in Section 2.5.2.*



**(d)** *In the context of the Example in Section 2.5.2, we see the sum of the multiplicity functions of vertices going from $v_0$ to the root $r_T$: $\varphi_T^{\Theta}(\{v_0\}) + \varphi_T^{\Theta}(\{v_0, v_{-1}\}) + \varphi_T^{\Theta}(r_T)$. The dotted lines represent critical points.*




**(e)** *The functions $f$ and $g$ in the Example in Section 2.5.3; with $\varepsilon = 0.3$.*



**(f)** *In the context of the Example in Section 2.5.3, we report $\varphi_T^{\Theta}(v_1)$ and $\varphi_T^{\Theta}(v_{-0.5})$. The dotted lines represent critical points.*

**Figure 2.6:** *Plots referring to the examples in Section 2.5.*

### 2.6.1 Decomposition Result

Since $d_E$ is topologically stable one can always suppose that a generalized dendrogram is given without order 2 vertices. Name $T_2$ the only representative without order 2 vertices inside the equivalence class of $T$. For notational convenience, from now on we suppose $T = T_2$ and $T' = T'_2$. To help us in the calculations define the following set of mappings: $\mathcal{M}(T, T') \subset Mapp(T, T')$ made by mappings $M$ such that $(v, G)$ or $(G, w)$ is in $M$ if and only if, respectively, $v \in V_T$ or $w \in V_{T'}$ is of order 2 after the deletions. The following Lemma then applies.

**Lemma 1.**

$$min\{cost(M)|M \in Mapp(T, T')\} = min\{cost(M)|M \in \mathcal{M}(T, T')\}$$

In addition to that, we consider some particular subsets of $E_T \times E_{T'}$ which play a fundamental role from now on. Recall that, using $E_T \simeq V_T - \{r_T\}$, we can induce $\pi_T : E_T \times E_{T'} \to V_T$.

A set $M^* \subset E_T \times E_{T'}$ is in $\mathcal{C}^*(T, T')$ if:

(A1) the points in $\pi_T(M^*)$ form antichains in $V_T$ (and the same for $\pi_{T'}(M^*)$ in $V_{T'}$), with respect to the partial order given by $father > son$. This means that any two distinct vertices of $T$ (respectively of $T'$) which appear in $M^*$ are incomparable with respect to "$>$";

(A2) the projections $\pi_T : M^* \to V_T$ and $\pi_{T'} : M^* \to V_{T'}$ are injective.

Consider now $M^* \in \mathcal{C}^*(T, T')$. Starting from such set of couples we build a set of edits which form a "partial" mapping between $T$ and $T'$. The meaning of "partial" will be made clear by Proposition 1. The main idea is that $M^*$ is used as a "dimensionality reduction tool": instead of considering the problem of finding directly the optimal mapping between $T$ and $T'$, we split up the problem in smaller subproblems, and put the pieces together using $M^*$. To do that, some other pieces of notation are needed.

Let $v \in E_T$. One can walk on the (unweighted) graph of the tree-structure of $T$ going towards any other vertex. For any $v \in E_T$, $\zeta_v$ is the shortest (in terms of vertices touched) graph-path connecting $v$ to $r_T$. Note that this is the ordered set $\zeta_v = \{v' \in V_T \big| v' > v\}$. Similarly, denote with $\zeta_x^{x'}$ the shortest path on the graph of $T$ connecting $x$ and $x'$. By Property (A1), given $x \in V_T \cap \pi_T(M^*)$, there exist $\widetilde{x} \neq x$ such that:

$$\widetilde{x} = min\{\cup_{\{(x',y') \in M^*, x' \neq x\}}(\zeta_x \cap \zeta_{x'})\}$$

And the same holds for $y \in V_{T'} \cap \pi_{T'}(M^*)$.

With these bits of notation, given $M^* \in \mathcal{C}^*(T, T')$, we now build the partial mapping $\alpha(M^*)$ with the following rules. Consider $v \in V_T$:

1. if $(v, w) \in M^*$, then $(v, w) \in \alpha(M^*)$;

2. if there is not $x \in V_T$ such that $v < \tilde{x}$ or $v > \tilde{x}$, then $(v, D) \in \alpha(M^*)$;
3. if there is $x \in V_T$ such that $v > \tilde{x}$ then $(v, D) \in \alpha(M^*)$;
4. if there is $x \in V_T$ such that $v < \tilde{x}$:
   (a) if $v \in \zeta_x^{\tilde{x}}$ then $(v, G) \in \alpha(M^*)$
   (b) if $v < v_i$ for some $v_i \in \zeta_x^{\tilde{x}} = \{v_0 < v_1 < \ldots < v_n\}$ then $(v, D) \in \alpha(M^*)$;
   (c) if $v < x$ no edit is associated to $v$.

**Remark 10.** *By Properties (A1) and (A2), the conditions used to build $\alpha(M^*)$ are mutually exclusive. This means that each $v \in V_T$ satisfies one and only one of the above conditions and so $\alpha(M^*)$ is well defined.*

The idea behind $\alpha(M^*)$ is that, for all couples $(x, y) \in M^*$, we want to turn $\zeta_x^{\tilde{x}}$ and $\zeta_y^{\tilde{y}}$ into single edges of the form $(x, \widetilde{x})$ and $(y, \tilde{y})$ respectively, and then shrink one in the other. Informally speaking, $\alpha(M^*)$ is a mapping that takes care of all the vertices in $T$ and $T'$, a part from the vertices $\cup_{(x,y)\in M^*}\{x' \in E_T \big| x' < x\}$ and $\cup_{(x,y)\in M^*}\{y' \in E_{T'} \big| y' < y\}$, that is the vertices which are below $x$, for $x \in V_T \cap \pi_T(M^*)$. For this reason we say that $\alpha(M^*)$ is a partial mapping.

Consider $T$ and $T'$ and $M^* \in \mathcal{C}^*(T, T')$. We obtain from such dendrograms, respectively, the dendrograms $\widetilde{T}_{M^*}$ and $\widetilde{T}'_{M^*}$ by deleting all the points $v < x$ and $w < y$ for each couple $(x, y) \in M^*$. With the following Proposition we give a formal description of the set of edits $\alpha(M^*)$.

**Proposition 1.** *The set $\alpha(M^*)$ is a mapping in $\mathcal{M}(\widetilde{T}_{M^*}, \widetilde{T}'_{M^*})$.*

Now we have all the pieces to obtain the following result.

**Theorem 2** (Decomposition)**.** *Given $T$, $T'$ dendrograms:*

$$d_E(T, T') = min_{M^*\in\mathcal{C}^*(T,T')} \sum_{(x,y)\in M^*} d_E(sub_T(x), sub_{T'}(y)) + cost(\alpha(M^*)) \quad (2.1)$$

### 2.6.2 Dynamical Integer Linear Programming problems

We want to use the Decomposition Theorem to write a dynamical, integer linear optimization algorithm to calculate $d_E$: by translating Theorem 2 into a Integer Linear Programming (ILP) problem, one obtains a single step in a bottom-up procedure. Here we give a concise description of the optimization problems involved in calculating $d_E$; a more detailed and comprehensive explanation can be found in the proof of Proposition 2.

### Notation

Consider $x \in V_T$ and $y \in V_{T'}$. Along with keeping the notation defined in Section 2.6.1, define $T_x := sub_T(x)$ and $T_y := sub_{T'}(y)$, $N_x := dim(T_x) = \#E_T$ and $N_y := dim(T_y) = \#E_{T'}$. In particular, given $v \in V_{T_x}$, the sequence $v_0 = v < v_1 < \ldots < r_T$ indicates the points in $\zeta_v$. The same with $w \in V_{T_y}$ .

### Setup and Variables

Suppose we already have $W_{xy}$ which is a $N_x \times N_y$ matrix such that $(W_{xy})_{v,w} = d_E(T_v, T_w)$ for all $v \in E_{T_x}$ and $w \in E_{T_y}$. Note that:

- if $x$ and $y$ are leaves, $W_{xy} = 0$.
- if $v, w$ are vertices of $T_x$, $T_y$, then $W_{vw}$ is a submatrix of $W_{xy}$.

The function to be optimized is defined on the following set of binary variables: for every $v \in E_{T_x}$ and $w \in E_{T_y}$, for $v_i \in \zeta_v$, $v_i < r_{T_x}$, and $w_j \in \zeta_w$, $w_j < r_{T_y}$, take a binary variable $\delta_{i,j}^{v,w}$. We write a constrained optimization problem such that having $\delta_{i,j}^{v,w} = 1$ means pairing the segments $\zeta_v^{v_{i+1}}$ (that is, the sequence of edges which starts with $(v, v_1)$ and ends with $(v_i, v_{i+1})$) and $\zeta_w^{w_{j+1}}$ , and shrinking one in the other in the induced mapping.

### Objective Function

Consider $v \in E_{T_x}$ and interpret $\delta_{i,j}^{v,w} = 1$ as coupling the segments $\zeta_v^{v_{i+1}}$ and $\zeta_w^{w_{j+1}}$; then $v$ is coupled with some $w \in E_{T_y}$ if $C(v) := \sum_{i,w,j} \delta_{i,j}^{v,w} = 1$ and is ghosted if $G(v) := \sum_{v' < v < v'_{i+1}, w, j} \delta_{i,j}^{v',w} = 1$. The vertex $v$ is instead deleted if $D(v) := 1 - C(v) - G(v) = 1$. We introduce also the following quantities, which correspond to the cost of shrinking $\zeta_v^{v_{i+1}}$ on $\zeta_w^{w_{j+1}}$:

$$\Delta_{i,j}^{v,w} = d(\oplus_{v' \in \zeta_v^{v_i}} \varphi_{T_x}(v'), \oplus_{w' \in \zeta_w^{w_j}} \varphi_{T_y}(w'))$$

Use $\delta$ to indicate the matrix of variables $(\delta_{i,j}^{v,w})_{v,w,i,j}$. The function which computes the cost given by coupled points is therefore:

$$F^C(\delta) := \sum_{v,w,i,j} \Delta_{i,j}^{v,w} \cdot \delta_{i,j}^{v,w}$$

The contribution of deleted points is: $F^D(\delta) - F^-(\delta)$, where

$$F^D(\delta) := \sum_{v \in T_x} D(v) \cdot d(\varphi_{T_x}(v), 0) + \sum_{w \in T_y} D(w) \cdot d(\varphi_{T_y}(w), 0)$$

and

$$F^{-}(\delta) := \sum_{v \in T_x} C(v) \cdot ||sub_{T_x}(v)|| + \sum_{w \in T_y} C(w) \cdot ||sub_{T_y}(w)||$$

where the "norm" of a tree $T$ is $||T|| = \sum_{e \in E_T} d(\varphi(e), 0)$.

Finally, one must take into account the values of $d_E(T_v, T_w)$, whenever $v$ and $w$ are coupled; this information is contained in $(W_{xy})_{v,w}$:

$$F^{S}(\delta) := \sum_{v,w} (W_{xy})_{v,w} \cdot \Big( \sum_{i,j} \delta_{i,j}^{v,w} \Big)$$

**Constraints**

The last piece of the equation is given by the constraints which must be satisfied by the variables $\delta_{i,j}^{v,w}$. For each $v' \in V_{T_x}$ we call $\Phi(v') := \{\delta_{i,j}^{v'',w} \big| v' = v_i'' \in \zeta_{v''}^{v_{i+1}''}\}$. In an analogous way we define $\delta(w')$ for $w' \in V_{T_y}$. Call $\mathcal{K}$ the set of values of $\delta$ such that for each leaf $l$ in $V_{T_x}$:

$$\sum_{v' \in \zeta_l} \sum_{\Phi(v')} \delta_{i,j}^{v'',w} \leqslant 1 \tag{2.2}$$

and for each leaf $l'$ in $V_{T_y}$:

$$\sum_{w' \in \zeta_{l'}} \sum_{\Phi(w')} \delta_{i,j}^{v,w''} \leqslant 1 \tag{2.3}$$

**Proposition 2.** *With the notation previously introduced:*

$$d_E(T_x, T_y) = min_{\delta \in \mathcal{K}} F^{C}(\delta) + F^{D}(\delta) - F^{-}(\delta) + F^{S}(\delta) \tag{2.4}$$

**Remark 11.** *A solution to Problem Equation* (2.4) *exists because the minimization domain is finite; it is not unique in general.*

## 2.7 Bottom-Up Algorithm

In this section the results obtained in Section 2.6 are used to obtain the algorithm implemented to compute the metric $d_E$ between generalized dendrograms. Some last pieces of notation are introduced in order to describe the "bottom-up" nature of the algorithm.

Given $x \in V_T$, define $len(x)$ to be the number of vertices in $\zeta_x$ and $len(T) = max_{v \in V_T} len(v)$. Therefore, $lvl(x) = len(T) - len(x)$. Lastly, $lvl_T(n) = \{v \in V_T | lvl(v) = n\}$

The key property is that: $lvl(x) > lvl(v)$ for any $v \in sub(x)$. Thus, if $W_{xy}$ is known for any $x \in lvl_T(n)$ and $y \in lvl_{T'}(m)$, then for any $v$, $w$ in $V_T$, $V_{T'}$ such that $lvl(v) < n$ and $lvl(w) < m$, $W_{vw}$ is known as well. With this notation we can write down Algorithm 2.1.

**Result:** $d_E(T, T')$
initialization: $N = len(T)$, $M = len(T')$, $n = m = 0$;
**while** $n \leqslant N$ *or* $m \leqslant M$ **do**
  **for** $(x, y) \in V_T \times V_{T'}$ *such that* $lvl(x) \leqslant n$ *and* $lvl(y) \leqslant m$ **do**
    Calculate $(W_{r_T r_{T'}})_{x,y}$ solving Problem (2.4);
  **end**
  $n = n + 1$; $m = m + 1$;
**end**
**return** $(W_{r_T r_{T'}})_{r_T, r_{T'}}$

**Algorithm 2.1:** Bottom-Up Algorithm.

We end up with a result to analyze the performances of Algorithm 2.1 in the case of dendrograms with binary tree structures.

**Proposition 3.** *Let $T$ and $T'$ be two generalized dendrograms with full binary tree structures with $dim(T) := \#E_T = N$ and $dim(T') = M$.*

*Then $d_E(T, T')$ can be computed bu solving $O(N \cdot M)$ ILP problems with $O(N \cdot log(N) \cdot M \cdot log(M))$ variables and $O(N + M)$ constraints.*

Note that binary dendrograms are dense (with respect to $d_E$) in any generalized dendrogram space as long as for any $\varepsilon > 0$, there is $x \in (X, \oplus, 0)$ such that $d(x, 0) < \varepsilon$. So this is indeed a quite general result.

### 2.7.1 Example

Here we present in details the first steps of the Algorithm 2.1, used to calculate the distance between two merge trees.

We consider the following couple of merge trees. Let $(T, h_T)$ be the merge tree given by: $V_T = \{a, b, c, d, r_T\}$, $E_T = \{(a, d), (b, d), (d, r_T), (c, r_T)\}$ and $w_T(a) = w_T(b) = w_T(d) = 1$, $w_T(c) = 5$; the merge tree $(T', h_{T'})$ instead, is defined by: $V_{T'} = \{a', b', c', d', r_{T'}\}$, $E_{T'} = \{(a', d'), (b', d'), (d', r_{T'}), (c', r_{T'})\}$ and $w_{T'}(a) = 1$, $w_{T'}(b) = w_{T'}(c) = 2$ and $w_{T'}(d) = 3$.

**Step:** $n = m = 0$

This step is trivial since we only have couples between leaves, like $(a, a')$, which have trivial subtrees and thus $d_E(sub_T(a), sub_{T'}(a')) = 0$.

**Step:** $n = m = 1$

The points $x \in V_T$ with $lvl_T(x) \leqslant 1$ are $\{a, b, c, d\}$ and the points $y \in V_{T'}$ with $lvl_{T'}(y) \leqslant 1$ are $\{a', b', c', d'\}$. Thus the couples $(x, y)$ which are considered are: $(d, d')$, $(d, a')$, $(d, b')$, $(d, c')$ and $(a, d')$, $(b, d')$, $(c, d')$. The couples between leaves, like $(a, a')$ have already been considered.

**Couple:** $(d, d')$ Let $T_d = sub_T(d)$ and $T_{d'} = sub_{T'}(d')$. The set of internal vertices are respectively $E_{T_d} = \{a, b\}$ and $E_{T_{d'}} = \{a', b'\}$. For each vertex $v <$ root in each subtree, where "root" stands for $d$ or $d'$, roots of $T_d$ and $T_{d'}$ respectively, we have $\zeta_v = \{v_0 = v, v_1 = \text{root}\}$. Thus, the binary variables we need to consider, are the following: $\delta_{0,0}^{a,a'}$, $\delta_{0,0}^{a,b'}$, $\delta_{0,0}^{b,a'}$ and $\delta_{0,0}^{b,b'}$. The quantities $\Delta_{i,j}^{v,w}$ are given by: $\Delta_{0,0}^{a,a'} = 0$, $\Delta_{0,0}^{a,b'} = 1$, $\Delta_{0,0}^{b,a'} = 0$ and $\Delta_{0,0}^{b,b'} = 1$. Thus:

$$F^C(\delta) = 0 \cdot \delta_{0,0}^{a,a'} + \delta_{0,0}^{a,b'} + 0 \cdot \delta_{0,0}^{b,a'} + \delta_{0,0}^{b,b'}$$

While:

$$F^D(\delta) = (1-\delta_{0,0}^{a,a'}-\delta_{0,0}^{a,b'})\cdot 1+(1-\delta_{0,0}^{b,a'}-\delta_{0,0}^{b,b'})\cdot 1+(1-\delta_{0,0}^{a,a'}-\delta_{0,0}^{b,a'})\cdot 1+(1-\delta_{0,0}^{a,b'}-\delta_{0,0}^{b,b'})\cdot 2$$

and:

$$F^-(\delta) = (\delta_{0,0}^{a,a'} + \delta_{0,0}^{a,b'}) \cdot 0 + (\delta_{0,0}^{b,a'} + \delta_{0,0}^{b,b'}) \cdot 0 + (\delta_{0,0}^{a,a'} + \delta_{0,0}^{b,a'}) \cdot 0 + (\delta_{0,0}^{a,b'} + \delta_{0,0}^{b,b'}) \cdot 0$$

and:

$$F^S(\delta) = \delta_{0,0}^{a,a'} \cdot 0 + \delta_{0,0}^{a,b'} \cdot 0 + \delta_{0,0}^{b,a'} \cdot 0 + \delta_{0,0}^{b,b'} \cdot 0$$

Lastly the constraints are:

$$\delta_{0,0}^{a,a'} + \delta_{0,0}^{a,b'} \leqslant 1;\ \delta_{0,0}^{b,a'} + \delta_{0,0}^{b,b'} \leqslant 1;\ \delta_{0,0}^{a,a'} + \delta_{0,0}^{b,a'} \leqslant 1;\ \delta_{0,0}^{a,b'} + \delta_{0,0}^{b,b'} \leqslant 1$$

A solution is given by $\delta_{0,0}^{a,a'} = \delta_{0,0}^{b,b'} = 1$ and $\delta_{0,0}^{a,b'} = \delta_{0,0}^{b,a'} = 0$, which entails $F^C(\delta) = 1$, $F^D(\delta) = 0$, $F^-(\delta) = 0$ and $F^S(\delta) = 0$ and $d_E(T_d, T_{d'}) = 1$.

**Couple:** $(d, a')$ Obviously: $d_E(sub_T(d), sub_{T'}(a')) = ||sub_T(d)||$. All the couples featuring a leaf and an internal vertex (that is, a vertex which is not a leaf), such as $(d, b')$, $(a, d')$ etc. behave similarly.

**Step:** $n = m = 2$

The points $x \in V_T$ with $lvl_T(x) \leqslant 2$ are $\{a, b, c, d, r_T\}$ and the points $y \in V_{T'}$ with $lvl_{T'}(y) \leqslant 2$ are $\{a', b', c', d', r_{T'}\}$. Thus the couples $(x, y)$ which are considered are $(d, r_{T'})$, $(r_T, d')$, $(r_T, r_{T'})$ and then the trivial ones: $(r_T, a')$, $(r_T, b')$, $(r_T, c')$ and $(a, r_{T'})$, $(b, r_{T'})$, $(c, r_{T'})$. Some couples have already been considered and thus are not repeated.

**Couple:** $(d, r_{T'})$ Let $T_d = sub_T(d)$ and $T' = sub_{T'}(r_{T'})$. The set of internal vertices are respectively $E_{T_d} = \{a, b\}$ and $E_{T_{d'}} = \{a', b', c', d'\}$. Thus, the binary variables we need to consider, are the following: $\delta_{0,0}^{a,a'}$, $\delta_{0,1}^{a,a'}$, $\delta_{0,0}^{a,b'}$, $\delta_{0,1}^{a,b'}$, $\delta_{0,0}^{a,c'}$, $\delta_{0,0}^{a,d'}$, $\delta_{0,0}^{b,a'}$, $\delta_{0,1}^{b,a'}$, $\delta_{0,0}^{b,b'}$, $\delta_{0,1}^{b,b'}$ $\delta_{0,0}^{b,c'}$, and $\delta_{0,0}^{b,d'}$.

The quantities $\Delta_{i,j}^{v,w}$ are given by: $\Delta_{0,0}^{a,a'} = 0$, $\Delta_{0,1}^{a,a'} = 3$, $\Delta_{0,0}^{a,b'} = 1$, $\Delta_{0,1}^{a,b'} = 4$, $\Delta_{0,0}^{a,c'} = 1$, $\Delta_{0,0}^{a,d'} = 2$, $\Delta_{0,0}^{b,a'} = 0$, $\Delta_{0,1}^{b,a'} = 3$, $\Delta_{0,0}^{b,b'} = 1$, $\Delta_{0,1}^{b,b'} = 4$, $\Delta_{0,0}^{b,c'} = 1$ and $\Delta_{0,0}^{b,d'} = 2$. The function $F^C(\delta)$ is easily obtained by summing over $\delta_{i,j}^{v,w} \cdot \Delta_{i,j}^{v,w}$.

While:

$$F^D(\delta) = (1 - \delta_{0,0}^{a,a'} - \delta_{0,1}^{a,a'} - \delta_{0,0}^{a,b'} - \delta_{0,1}^{a,b'} - \delta_{0,0}^{a,c'} - \delta_{0,0}^{a,d'}) \cdot 1 + \ldots + (1 - \delta_{0,0}^{a,d'} - \delta_{0,0}^{b,d'}) \cdot 3$$

and:

$$F^-(\delta) = (\delta_{0,0}^{a,a'} + \delta_{0,1}^{a,a'} + \delta_{0,0}^{a,b'} + \delta_{0,1}^{a,b'} + \delta_{0,0}^{a,c'} + \delta_{0,0}^{a,d'}) \cdot 0 + \ldots + (\delta_{0,0}^{a,d'} + \delta_{0,0}^{b,d'}) \cdot 3$$

and:

$$F^S(\delta) = (\delta_{0,0}^{a,a'} + \delta_{0,1}^{a,a'}) \cdot 0 + (\delta_{0,0}^{a,b'} + \delta_{0,1}^{a,b'}) \cdot 0 + \ldots + \delta_{0,0}^{a,d'} \cdot 3 + \delta_{0,0}^{b,d'} \cdot 3$$

Lastly the constraints are:

$$\delta_{0,0}^{a,a'} + \delta_{0,1}^{a,a'} + \delta_{0,0}^{a,b'} + \delta_{0,1}^{a,b'} + \delta_{0,0}^{a,c'} + \delta_{0,0}^{a,d'} \leqslant 1$$

$$\delta_{0,0}^{b,a'} + \delta_{0,1}^{b,a'} + \delta_{0,0}^{b,b'} + \delta_{0,1}^{b,b'} + \delta_{0,0}^{b,c'} + \delta_{0,0}^{b,d'} \leqslant 1$$

$$\delta_{0,0}^{a,a'} + \delta_{0,1}^{a,a'} + \delta_{0,0}^{b,a'} + \delta_{0,1}^{b,a'} + \delta_{0,0}^{a,d'} + \delta_{0,0}^{b,d'} \leqslant 1$$

$$\delta_{0,0}^{a,b'} + \delta_{0,1}^{a,b'} + \delta_{0,0}^{b,b'} + \delta_{0,1}^{b,b'} + \delta_{0,0}^{a,d'} + \delta_{0,0}^{b,d'} \leqslant 1$$

$$\delta_{0,0}^{a,c'} + \delta_{0,0}^{b,c'} \leqslant 1$$

In this case there are many minimizing solutions. One is given by: $\delta_{0,1}^{a,a'} = \delta_{0,0}^{b,c'} = 1$ and all other variables equal to $0$. This value of $\delta$ is feasible since the variables $\delta_{0,1}^{a,a'}$ and $\delta_{0,0}^{b,c'}$ never appear in the same constraint. This value of $\delta$ entails $F^C(\delta) = 3 + 1$, $F^D(\delta) = 2$, $F^-(\delta) = 0$ and $F^S(\delta) = 0$, and thus $d_E(T_d, T') = 6$.

Another solution can be obtained with: $\delta_{0,0}^{a,d'} = \delta_{0,0}^{b,c'} = 1$ and all other variables equal to $0$. Also this value of $\delta$ is feasible since the variables $\delta_{0,0}^{a,d'}$ and $\delta_{0,0}^{b,c'}$ never appear in the same constraint. This value of $\delta$ entails $F^C(\delta) = 2+1$, $F^D(\delta) = w_{T'}(a') + w_{T'}(b') = 1 + 2$, $F^-(\delta) = ||sub_{T'}(d')|| = 3$ and $F^S(\delta) = d_E(sub_T(a), sub_{T'}(d')) = ||sub_{T'}(d')|| = 3$, and thus $d_E(T_d, T') = 3 + 3 - 3 + 3 = 6$.

**Couple:** $(r_T, d')$ This and the other couples are left to the reader.

## 2.8 Numerical Simulations

In this section, the feasibility of the algorithm presented in Section 2.7 is assessed by means of some numerical simulations. We also deal with the problem of approximating the metric $d_E$ when the number of leaves in the tree structures in the data set is too big to be handled. The effectiveness of such approximations is showcased using some simple case studies, which also give some practical examples of the issues raised in Section 2.2. In the implementations, dendrograms are always considered with a binary tree structure, obtained by adding negligible edges, that is edges $e$ with arbitrary small $d(\varphi(e), 0)$, when the number of children of a vertex exceeds 2.

### 2.8.1 Edit Distance Simulations

To get some concrete ideas of proper runtimes needed to calculate distances, we fix the number of leaves $n$ and for $100$ times the following procedure is repeated: generate two random samples of $n$ points from the uniform distribution on a compact, real interval, take their single linkage hierarchical dendrograms (with multiplicity function equal to the weight function $w_T$) and compare them with $d_E$. This whole pipeline is repeated for any integer $n$ in the interval $[5, 20]$. In Figure 2.7 there are the average runtimes as a function of the number of leaves of the involved binary trees. The standard deviations over the repetitions are also reported, which show a quite large band around the mean. The different curves in Figure 2.7 concern the portion of time effectively spent by the solver to compute the solution of the ILP problems, and the amount of time employed to setup such problems. All code is written in Python and thus this second part of the runtimes can likely be greatly reduced by using more performing programming languages. The green line of total time is computed parallelizing the **for** loop in Algorithm 2.1. Note that dendrograms with the same number of leaves may end up having different tree-structures and so different dimensions. This is the main reason for the big shaded regions around the mean. If the trees were aggregated by dimension, the standard deviation of runtimes would decrease. Nevertheless, in applications, the only thing one can reasonably control is the number of leaves (which is given by the number of minima in the function, the number of clusters in a dendrogram, etc.) and for this reason the trees are aggregated as in Figure 2.7.

The computations are carried out on a 2016 laptop with Intel(R) processor Core(TM) i7-6700HQ CPU @ 2.60GHz, 4 cores (8 logical) and 16 GB of RAM. The employed ILP solver is the freely available IBM CPLEX Optimization Studio 12.9.0.

### 2.8.2 Pruning

In Section 2.2.1 we point out that the merging information carried by the dendrograms in Figure 2.1 is mostly contained in a number of vertices which is much lower than the actual number of leaves in the dendrogram. Another perspective on the same fact

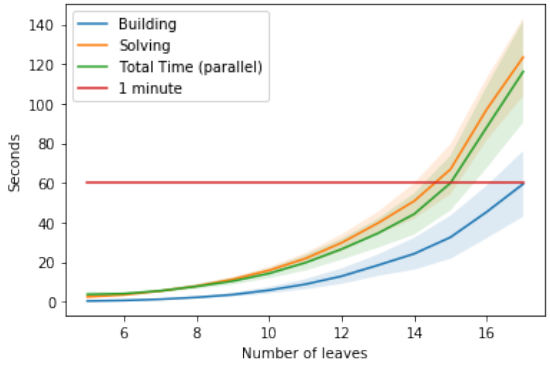


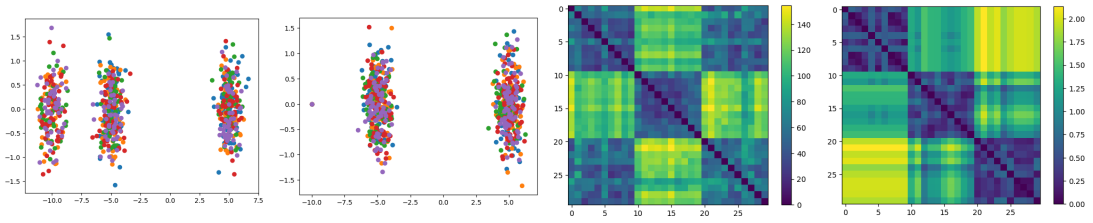

**Figure 2.7:** *Graph of the computational times as function of the number of leaves. The curves represent running times to calculated $d_E$ between couples of merge trees, averaged over* 100 *random couples of trees, with shaded regions including intervals of $+/-$ one standard deviation. "Building time" means the time spent by Python to setup the ILP problems. "Solving time" is the time used by the solver to solve the LIP problems. "Total time" is the time spent computing the distance using parallel computing of the ILP problems: both for the building and solving steps.*

is that, if one defines a proper multiplicity function, with values in an editable space $X$, coherently with the aim of the analysis, then the value $d(\varphi(e), 0)$ can be thought as the amount of information carried by the edge $e$. The bigger such value is, the more important that edge will be for the dendrogram. In fact such edges are the ones most relevant in terms of $d_E$.

A sensible way to reduce the computational complexity of the metric $d_E$, losing as little information as possible, is therefore the following: consider any couple of leaves, if the "amount of information" $d(\varphi(e), 0)$ of one of the two leaves is below a certain threshold, that leaf is deleted and its father ghosted. If both are below the threshold, only the leaf with smaller $d(\varphi(e), 0)$ is deleted (if $d(\varphi(e), 0)$ is equal across two siblings, one of them is randomly deleted). This operation is repeated recursively until no leaf with multiplicity under the fixed threshold is left. This operation is called pruning; the operator which assigns to a dendrogram the pruned dendrogram with $\varepsilon \geqslant 0$ threshold is called $P_\varepsilon$. Note that pruning a dendrogram removes leaves, but keeps unchanged the distance from the root to the leaves which are not deleted. For instance, in the case of merge trees, this means that the range of the height function $h_T$ does not change upon pruning the tree. We can quantify the (normalized) lost information with what we call *pruning error* ($PE$): $(||T|| - ||P_\varepsilon(T)||)/||T||$.

### 2.8.3 Examples

Now we use two simulated data sets to put to work the frameworks defined in Section 2.5. The examples are basic, but suited to assert that generalized dendrograms and



**(a)** *Data from the second class.* **(b)** *Data from the third class.* **(c)** *Pairwise distances with dendrograms.* **(d)** *Pairwise distances with PD.*

**Figure 2.8:** *Data and pairwise distance matrices involved in the hierarchical clustering example.*

the metric $d_E$ capture the information we designed them to grasp. In particular, since examples in Section 2.2.1 and Section 2.2.2 already give insights into the role of the tree-structured information, we want to isolate and emphasize the key role of multiplicity functions. The examples presented concern hierarchical clustering dendrograms and dendrograms representing scalar fields.

### Hierarchical Clustering Dendrograms

We consider a data set of $30$ points clouds in $\mathbb{R}^2$, each with $150$ or $151$ points. Point clouds are generated according to three different processes and are accordingly divided into three classes. Each of the first $10$ point clouds is obtained by sampling independently two clusters of $75$ points respectively from normal distributions centered in $(5, 0)$ and $(-5, 0)$, both with $0.5 \cdot Id_{2\times 2}$ covariance. Each of the subsequent $10$ point clouds is obtained by sampling independently $50$ points from each of the following Gaussian distributions: one centered in $(5, 0)$, one in $(-5, 0)$ and one in $(-10, 0)$. All with covariance $0.5 \cdot Id_{2\times 2}$. Lastly, to obtain each of the last $10$ point clouds, we sample independently $150$ points as done for the first $10$ clouds, that is $75$ independent samples from a Gaussian centered $(5, 0)$ and $75$ from one centered in $(-5, 0)$, an then, to such samples, we add an outlier placed in $(-10, 0)$.

Some clouds belonging to the second class and third classes are plotted respectively in Figure 2.8(a) and Figure 2.8(b). We resort to pruning because of the high number of leaves, but we still expect to be able to easily separate point clouds belonging to the first and third classes (that is, with two major clusters) from clouds belonging to the second class, which feature three clusters, thanks to the cardinality information function defined in Section 2.5.2. All dendrograms have been pruned with the same threshold, giving an average pruning error of $0.15$.

We can see in Figure 2.8(c) that this indeed the case. It is also no surprise that persistence diagrams do not perform equally good in this classification task, as displayed in Figure 2.8(d). In fact PDs have no information about the importance of the cluster, making it impossible to properly recognize the similarity between data from the first and

third class. They are, however, able to distinguish clouds belonging to class two from clouds belonging to class three since the persistence of the homology class associated to the leftmost cluster in clouds belonging to class two is smaller compared to what happens in clouds from the third class. The cluster centered in $-10$ and the one in $-5$ are in fact closer when the first one is a proper cloud, than when it is a cluster made by a single point.

#### Dendrograms of Functions

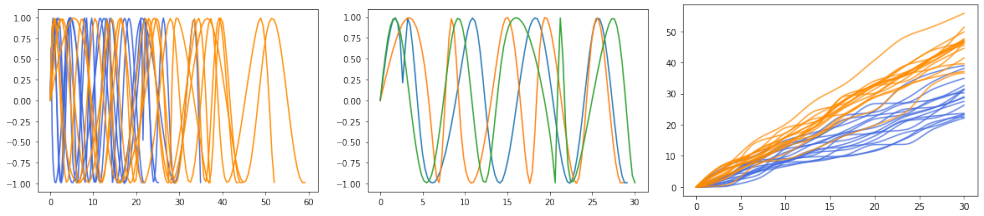
This time our aim is to work with merge trees of functions, adding the multiplicity function induced by the Lebesgue measure of the sublevel sets, as defined in Section 2.5.3, and using them to discriminate between two classes in a functional data set.

We simulate the data set so that the discriminative information is contained in the size of the sublevel sets and not in the structure of the critical points . To do so a situation which is very similar to the one shown by Sangalli et al. (2010) for the Berkeley Growth Study data is reproduced, where all the variability between groups in a classification task is explained by warping functions. We fix a sine function defined over a compact $1D$ real interval (with the Lebesgue measure) and we apply to its domain $100$ random non linear warping functions belonging to two different, but balanced, groups. Warpings from the first group are more likely to obtain smaller sublevel sets, while in the second groups we should see larger sublevel sets and so "bigger" multiplicity functions defined on the edges. Note that, being the Lebesgue measure invariant with the translation of sets, any horizontal shifting of the functions would not change the distances between dendrograms.

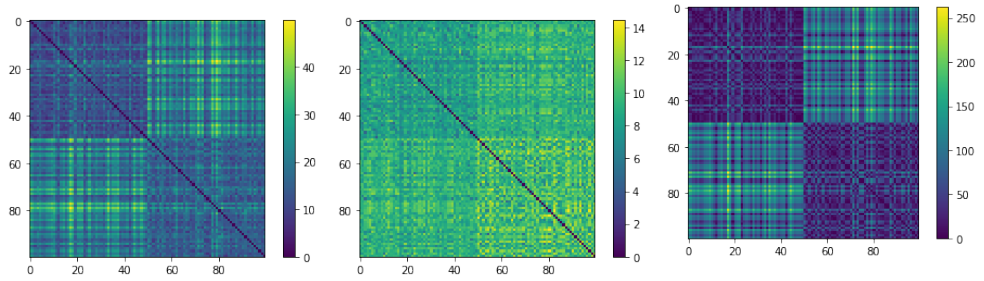
The base interval is $I = [0, 30]$ and the base function is $f(x) = \sin(x)$. The warping functions are drawn in the following way. Pick $N$ equispaced control points in $I$ and then we draw $N$ samples from a Gaussian distribution truncated to obtain only positive values. We thus have $x_1, \dots x_N$ control points and $v_1, \dots, v_N$ random positive numbers. Define $y_i := \sum_{j=1}^{i} v_j$. The warping is then obtained interpolating with monotone cubic splines the couples $(x_i, y_i)$. Being the analysis invariant to horizontal shifts in the functions, we fix $x_0 = y_0 = 0$ for visualization purposes.

The groups are discriminated by the parameters of the Gaussian distribution from which we sample the positive values $v_i$ to set up the warping. For the first class we sample $N = 10$ positive numbers from a truncated Gaussian with mean $3$ and standard deviation $2$; for the second the mean of the Gaussian is $5$ and the standard deviation is $2$. Thus we obtain each of the first $50$ functions sampling $10$ values $v_i$ from the truncated Gaussian centered in $3$, building the warping function as explained in the previous lines, and then reparametrizing the sine function accordingly. The following $50$ functions are obtained with the same pipeline but employing a Gaussian centered in $5$. Note that, by construction, all the functions in the data set share the same merge tree.

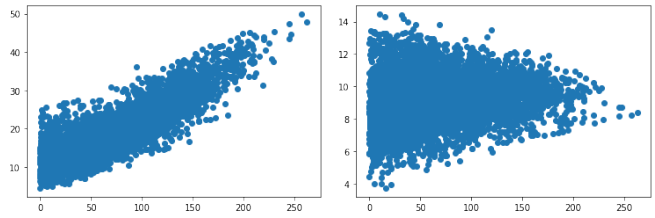
Examples of the warping functions can be seen in Figure 2.9(c); the resulting functions can be seen in Figure 2.9(a). The key point here is that we want to see if the



**(a)** *Subset of functions colored by classes.* **(b)** *Few functions from the first class.* **(c)** *Subset of warping functions colored by classes.*



**(d)** *Pairwise distances with dendrograms.* **(e)** *Pairwise distances with $L_2$ metric.* **(f)** *Pairwise distances with $L_2$ metric on warping functions.*



**(g)** *Correlation between dendrograms and warping functions metric.* **(h)** *Correlation between naive $L_2$ and warping functions metric.*

**Figure 2.9:** *Overview of the example of Section 2.8.3.*

dendrograms can retrieve the information contained in the warping functions. For this reason we compare the $L_2$ pairwise distances between such functions (see Figure 2.9(f)) and the pairwise distances obtained with dendrograms (see Figure 2.9(d)). The visual inspection confirms the close relationships between the two sources of information. Moreover, if we vectorize the arrays given by the two matrices (considering only entries above the diagonal) and compute the Fisher correlation, we get a score of $0.85$ (see Figure 2.9(g)). Instead, a naive approach with the $L_2$ metric applied directly to the data set would capture no information at all, as we can observe from Figure 2.9(e) and the Fisher correlation with the matrix obtained from warping functions is $0.15$ (see Figure 2.9(h)).

Note that, in general, the problem of finding warping functions to align functional data is deeply studied and with no easy solution (see, for instance, the special issue of the Electronic Journal of Statistics dedicated to phase and amplitude variability - year 2014, volume 8 or Srivastava et al. (2011)) especially for non-linear warping of multidimensional or non-euclidean domains. Instead, generalized dendrograms less sensitive to such dimensionality issues, in the sense that they only arise in calculating the connected components and measure of the sublevel sets.

## 2.9 Conclusions

Starting from the problems outlined in Section 2.2, we develop a framework to work with topological information which can be represented with tree-like structures. As motivated throughout the manuscript, we argue that these kinds of topological summaries can succeed in many situations where persistence diagrams are not effective. They also provide a great level of versatility because of the wide range of additional information that can be extracted from data. Possibly the greatest drawback in these representations is the computational complexity involved in comparing them. We define a novel metric structure which works locally on the trees and can be calculated by solving a set of smaller and easier subproblems. This metric proves to be feasible and we carry out some examples to showcase its effectiveness in situations which are of interest in different branches of data analysis.

Along with the more general perspective of finding other ways to enrich the information extracted by TDA from data, this work leaves many paths that can be followed. We think that the hypotheses on the set of weights which can be added to dendrograms can be relaxed; however, the algorithm presented in this manuscript may need to be adapted to the properties of the chosen weight space. Even without increasing the range of available weights, case-specific pipelines can be designed and studied, as done in the case of merge trees of functions in Chapter 3. Moreover interaction with the more general case of Reeb Graphs can be investigated, possibly following the decomposition presented in Stefanou (2020). Lastly other families of metrics can be defined, starting from $d_E$, aiming at emphasizing or overlooking on certain kinds of variability in the

dendrograms, providing more "stable" metrics.

## 2.10 Proofs

*Proof of* Theorem 1.

To lighten the notation we use the following symbols:

- the edit induced by $(v, D)$ is called $v_d$ and $v_d^{-1}$ stands for $(D, v)$.
- the edit induced by $(v, G)$ is called $v_g$ and $v_g^{-1}$ stands for $(G, v)$.
- the edit induced by $(v, v')$ is called $v_{\varphi,\varphi'}$ with $\varphi$ being the original multiplicity function, and $\varphi'$ the multiplicity function after the shrinking.

We know that the set of finite edit paths between two dendrograms is nonempty.

Suppose that $\gamma$ is a finite edit path. This means that $\gamma$ is the composition of a finite set of edits. We indicate such ordered composition with $\gamma = \prod_{i=0}^{N} e_i$ with $e_i$ edit operation. We would like to change the order of the edit operations without raising the cost and changing the extremes of the edit path. This is not always possible. However we can work it around in the useful cases using properties (P1)-(P4). In particular, we would like to know when we can commute a generic edit $e_i$ in the following situations:

- $v_d \circ e_i$ and $e_i \circ v_d^{-1}$
- $v_g \circ e_i$ and $e_i \circ v_g^{-1}$.

Moreover we want to reduce the edit path to max one edit for any vertex of $T$ and $T'$.

We divide the upcoming part of the proof in subsections, each devoted to different combinations of edits.

### $v_d$ and $v_d^{-1}$

When we delete or insert one vertex, we are modifying the tree structure at the level of its father and its children. Therefore, we are only taking into account operations on the father, on the vertex himself or on the children of the deleted/inserted vertex.

- $v_d \circ v'_g$ with $v$ son of $v'$, can be safely replaced with $v_d \circ v'_d$. Instead of ghosting the father and then deleting the whole edge, we can delete both edges one by one; conserving the length of the path (P3). If $v$ is father of $v'$ then we can safely commute the operations.

- $v_d \circ v_g^{-1}$ can be replaced with $v'_{\varphi,\varphi'}$ with $v'$ father of $v$ (after the insertion) and $\varphi'$ properly defined not to raise the cost of the path. In fact we are inserting $v$ on an edge and then deleting it. This can obviously be achieved by shrinking the original edge (without changing the path length - (P4)).
- similarly, $v_d \circ v_g'^{-1}$ with $v'$ to be father of $v$ can again be replaced safely by a proper shrinking: instead of inserting a point in an edge, and deleting then the edge below, we can directly shrink the original edge (P4). If $v'$ is to be inserted below $v$ this is the same situation, but seen from the point of view of the son of $v$.
- $v_d \circ v_{\varphi,\varphi'}$ can be replaced by $v_d$ potentially diminishing the length of the path, but surely not raising it (P1).
- $v'_g \circ v_d^{-1}$. If $v'$ is the father of $v$, this edit can be replaced with just $v'_{\varphi,\varphi'}$ with appropriate weights: we are inserting an edge under a vertex which (in this case) becomes of order two and is ghosted. We can directly modify the edge without changing the length of the path (P4). If $v'$ is the vertex which would become son of $v$, we can simply shrink $v$ to obtain the same result without raising the cost (P4).
- $v_g'^{-1} \circ v_d^{-1}$, with $v'$ to appear on the edge inserted with $v_d^{-1}$ cannot commute (otherwise can always commute), but can be replaced by two insertions: instead of inserting an edge and then splitting it, we can directly insert two smaller edges; without changing the cost of the path (P3).
- $v_{\varphi,\varphi'} \circ v_d^{-1}$ can be replaced with an insertion directly with multiplicity $\varphi'$, possibly shortening the path (P1).
- consider $v_d'^{-1} \circ v_d$ with $v'$ to be inserted with, as father, the father of $v$; if the children of $v'$ are different from the children of $v$, this operation cannot commute. If the children are the same, it can be changed with a shrinking of $v$, reducing the length of the path by at most $cost(v_d'^{-1}) + cost(v_d)$ (P1).

**$v_g$ and $v_g^{-1}$**

Like in the previous case, we only take into account transformations concerning the father and the son of the added/ghosted order two vertex.

- $v_g \circ v'_g$, with $v$ and $v'$ being on adjacent edges, can commute (P2).
- $v_g \circ v_g'^{-1}$, with $v$ and $v'$ being on adjacent edges, can commute provided we define carefully the splitting $v_g'^{-1}$ (P2).
- $v_g \circ v_{\varphi,\varphi'}$ means that we are shrinking a vertex before ghosting it. However, we can achieve the same result, without increasing the path length, by ghosting the vertex at first, and then shrinking its son (P1)-(P4).

- $v'_{\varphi,\varphi'} \circ v_g^{-1}$ either with $v' = v$, or with $v$ father of $v'$, can be replaced with an appropriate shrinking of the (future) son of $v$, and then an appropriate insertion of $v'$ without changing the length of the path (P3)-(P4).
- $v_g \circ v'_d$ with $v$ father of $v'$ cannot be commuted and cannot be replaced by a similar operation which inverts ghosting and deletion.

$v_{w,w'}$

- $v_{\varphi',\varphi''} \circ v_{\varphi,\varphi'}$ can be replaced by $v_{\varphi,\varphi''}$ which is either conserving or shortening the path (P1).
- $v_{\varphi,\varphi'}^{-1} = v_{\varphi',\varphi}$.

Thanks to these properties we can take a given path $\gamma = \prod_{i=0,\dots,N} e_i$ and modify the edit operations in order to obtain the following situation:

- the first operations are all in the form $v_d$; this can be achieved because $v_d \circ -$ can be always rearranged, potentially by changing the path as shown before and shortening it. Of course there can be only one deletion for each vertex of $T$;
- then we have all the edits in the form $v_g$; since $v_g \circ -$ is exchangeable any time but when we have $v_g \circ v'_d$, this is not a problem. Observe that all order two vertices which were not deleted can be ghosted (at most one time);
- in the same way we can put last all the paths in the form $v_d^{-1}$ and before them $v_g^{-1}$. All the new vertices appearing with the insertion of edges and the splitting of edges with order two vertices are all nodes which remain in $T'$ and which are not further edited;
- in the middle we are left with the shrinking paths. Since we can substitute $v_{\varphi,\varphi'} \circ v_{\varphi',\varphi''}$ with $v_{\varphi,\varphi''}$, we can obtain just one single transformation on a vertex.

Thus

$$\overline{\gamma} = (\gamma_d^{T'})^{-1} \circ (\gamma_g^{T'})^{-1} \gamma_s^T \circ \gamma_g^T \circ \gamma_d^T .$$

with:

- $\gamma_d^T = \prod v_d$
- $\gamma_g^T = \prod v_g$
- $\gamma_s^T = \prod v_{\varphi,\varphi'}$
- $(\gamma_g^{T'})^{-1} = \prod v_g^{-1}$
- $(\gamma_d^{T'})^{-1} = \prod v_d^{-1}$

is such that $\gamma(T) = \overline{\gamma}(T) = T'$ and $cost(\overline{\gamma}) \leqslant cost(\gamma)$. The key point is that $\overline{\gamma}$ can be easily realized as a mapping in the following way:

- $(v, D)$ $\forall v_d \in \gamma_d^T$
- $(v, G)$ $\forall v_g \in \gamma_g^T$
- $(v, v')$ $\forall v_{\varphi,\varphi'} \in \gamma_s^T$, where $v'$ is the renaming of $v$, with multiplicity given by $\varphi'$.
- $(G, v)$ $\forall v_g^{-1} \in (\gamma_g^{T'})^{-1}$
- $(D, v)$ $\forall v_d^{-1} \in (\gamma_d^{T'})^{-1}$

■

*Proof of* Lemma 1.

Any order 2 vertex which is not ghosted is paired with another order 2 vertex. Ghosting both of them does not increase the cost of the mapping.

■

*Proof of* Proposition 1.

Condition (M2) coincide with condition (A2). Condition (M3) is clearly satisfied because of the antichain condition (A1). Consider a vertex $v \in E_T$. The only case in which $v$ is not edited is when $v < x$ with $x \in v_T \cap \pi_T(M^*)$. However, in this case $v$ does not appear in $\widetilde{T}_{M^*}$, and thus (M1) is satisfied. Moreover, all and only order 2 vertices, after the deletions, are ghosted, and (M4) follows .

■

*Proof of* Theorem 2.

Let $M \in \mathcal{M}(T, T')$ such that $d_E(T, T') = cost(M)$.

We note that $father > son$ induces a partial order relationship also on the pairs given by coupled points in $M$: $(x, y) > (v, w)$ if $x > v$ and $y > w$. In fact, by property (M3), $x > v$ if and only if $y > w$. So we can select $(x_i, y_i)$, the maxima with respect to this partial order relationship. Thus, we obtain $(x_0, y_0)$,...,$(x_n, y_n)$ which form an antichain (both in $V_T$ and $V_{T'}$).

Clearly $M^* = \{(x_0, y_0),...,(x_n, y_n)\} \in \mathcal{C}^*(T, T')$. Now we build $\alpha(M^*)$ and compare the cost of its edits with the ones in $M$. Let $\bar{x}$ be the minimal common parent between $x_i$ and $x_j$. Since $\bar{x} > x_i, x_j$, it is not coupled in $M$. Since $x_i$ and $x_j$ are coupled, $\bar{x}$ cannot be ghosted, so it is deleted in $M$. Any point $x$ above $\bar{x}$ is deleted for the same reasons. So the edits above $\bar{x}$ are shared between $\alpha(M^*)$ and $M$.

In $\alpha(M^*)$ we ghost any point between $\bar{x}$ and $x_i$ (and the same for $x_j$) and this is not certain to happen in $M$ (some points could be deleted). Nevertheless, even in the worst

case, these ghostings are guaranteed not to increase the distance. For instance, suppose $x_i < x < \bar{x}$ is deleted in $M$ and ghosted by $\alpha(M^*)$, then:

$$d(x_i \oplus x, y_i) \leqslant d(x_i \oplus x, y_i \oplus x) + d(y_i \oplus x, y_i) = d(x_i, y_i) + d(x, 0)$$

by properties (P1)-(P4). Since $\alpha(M^*) \in \mathcal{M}(\widetilde{T}_{M^*}, \widetilde{T}'_{M^*})$ by Proposition 1, we have:

$$\sum_{(x,y)\in M^*} d_E(sub_T(x), sub_{T'}(y)) + cost(\alpha(M^*)) \leqslant cost(M)$$

Now we prove the other inequality.

Consider $M^*$ which realizes the minimum of the right side of Equation (2.1), and $M_i$ which realizes $d_E(sub(x_i), sub(y_i))$ with $(x_i, y_i) \in M^*$. We build a mapping $M$ collecting edits in the following way: for every $x' \in E_T$ if $x' \in sub(x_i)$, we take the edit associated to it from $M_i$, otherwise we know that it is edited by $\alpha(M^*)$, and we take it from there; the set of these assignments gives $M \in \mathcal{M}(T, T')$ whose cost is exactly $\sum_{(x_i,y_i)\in M^*} cost(M_i) + cost(\alpha(M^*))$. This gives the second inequality.

■

*Proof of* Proposition 2.

We use all pieces of notation defined in Section 2.6.1. Consider $x \in E_T$ and $y \in E_{T'}$. Recall that $T_x := sub_T(x)$ and $T_y := sub_{T'}(y)$, and $N_x := dim(T_x)$ and $N_y := dim(T_y)$. Moreover, given $v \in V_{T_x}$, we use the sequence $v_0 = v < v_1 < \ldots < r_T$ to indicate the points in $\zeta_v$. The same with $w \in V_{T_y}$ .

**Setup**

We have $W_{xy}$ which is a $(N_x - 1) \times (N_y - 1)$ matrix such that $(W_{xy})_{v,w} = d_E(T_v, T_w)$ for all $v \neq x \in T_x$ and $w \neq y \in T_y$.

We would like to find $M^* \in \mathcal{C}^*(T_x, T'_y)$ minimizing Equation (2.1) for $T_x$ and $T_y$, but this is a difficult task. In fact, as evident in the construction of $\alpha(M^*)$, a set $M^* \in \mathcal{C}^*(T_x, T_y)$ has the role of pairing segments of dendrograms: if $(v, w) \in M^*$, then the paths $\zeta_v^{\tilde{v}}$ and $\zeta_w^{\tilde{w}}$ are paired and then shrunk one on the other by $\alpha(M^*)$. But the points $\tilde{v}$ and $\tilde{w}$ depend on the whole set $M^*$, and not simply on the couple $(v, w)$. Modeling such global dependence gives rise to non-linear relationships between coupled points, and so leading to a non linear cost function, in terms of points interactions, to be minimized. For this reason we "weaken" the last term in Equation (2.1), allowing also mappings different from $\alpha(M^*)$ to be built from $M^*$. In other words we minimize the following equation:

$$\sum_{(x,y)\in M^*} d_E(sub_T(x), sub_{T'}(y)) + cost(\beta(M^*)) \tag{2.5}$$

where $M^*$ is the set of coupled points in $\beta(M^*)$ and lies in $\mathcal{C} * (T, T')$, and $\beta(M^*)$ is a mapping in $\mathcal{M}(\widetilde{T}_{M^*}, \widetilde{T}'_{M^*})$. Since $\alpha(M^*)$ fits these conditions, minimizing Equation (2.1) or Equation (2.5) gives the same result.

### Variables

As already stated, we are considering the following set of binary variables: for every $v \in E_{T_x}$ and $w \in E_{T_y}$, for $v_i \in \zeta_v$, $v_i < r_{T_x}$, and $w_j \in \zeta_w$, $w_j < r_{T_y}$, we have a binary variable $\delta_{i,j}^{v,w}$. We want to write a constrained optimization problem such that having $\delta_{i,j}^{v,w} = 1$ means that we pair the segments $\zeta_v^{v_{i+1}}$ and $\zeta_w^{w_{j+1}}$, and shrink one in the other in the induced mapping. This, for instance implies that $(v, w) \in M^*$ and, when designing the constraints, $v_{i+1}$ (and the same for $w_{j+1}$) is the first point going from $v$ towards $r_{T_x}$, such that there can be another point in $sub_{T_x}(v_{i+1})$ paired in $\beta(M^*)$.

Now we state how $\delta_{i,j}^{v,w} = 1$ contributes to define edits in $\beta(M^*)$, which is then given by the edits induced by all the variables equal to 1. In order to pair and shrink the segments $\zeta_v^{v_{i+1}} = \{v = v_0, v_1, \dots, v_{i+1}\}$ and $\zeta_w^{w_{j+1}}$ we need to induce the following edits on $T_x$:

- all the points $v_k \in \zeta_v^{v_{i+1}}$ with $0 < k < i + 1$ are ghosted, that is $(v_k, G) \in \beta(M^*)$;
- if $v' < v_k$ for some $0 < k < i + 1$, then $(v', D) \in \beta(M^*)$;
- if $v' \geqslant v_{i+1}$ and $v' \neq r_T$, then $(v', D) \in \beta(M^*)$
- $(v, w) \in \beta(M^*)$.

Of course analogous edits must be induced on points in $T_y$. Thus, the edit $(v, w) \in \beta(M^*)$, in the edit paths given by the mapping $\beta(M^*)$, means shrinking the edge $(v, v_{i+1})$ onto $(w, w_{j+1})$). Recall that, if $\delta_{i,j}^{v,w} = 1$, we do not need to define edits for $sub_{T_x}(v)$ and $sub_{T_y}(w)$ since, by assumption, we already know $d_E(T_v, T_w)$.

### Constraints

Clearly, not all combinations of $\delta_{i,j}^{v,w}$ are acceptable, in that they do not induce a proper partial mapping $\beta(M^*)$, with $M^* \in \mathcal{C}^*(T_x, T_y)$. Segments could even be paired multiple times. To avoid such issues, we build a set of constraints for the variable $\delta$. Recall that the set of acceptable values $\mathcal{K}$ is defined by Equation (2.2) and Equation (2.3).

The following Proposition clarifies the properties of any value of $\delta \in \mathcal{K}$.

**Proposition 4.** *If $\delta \in \mathcal{K}$:*

- *the couples $(v, w)$ such that $\delta_{i,j}^{v,w} = 1$ define a set $M^* \in \mathcal{C}^*(T_x, T_y)$;*
- *the edits induced by all $\delta_{i,j}^{v,w} = 1$ give a mapping $\beta(M^*)$ in $\mathcal{M}(\widetilde{T}_{xM^*}, \widetilde{T}_{yM^*})$.*

**Remark 12.** *If for every $\delta_{i,j}^{v,w} = 1$, $v_{i+1} = \tilde{v}$, then $\beta(M^*) = \alpha(M^*)$.*

**Objective Function**

Now we build a linear cost functions which calculates the results of Equation (2.5) for $\delta \in \mathcal{K}$. The key point we need to address is how to calculate the cost of $\beta(M^*)$.

Consider $v \in E_{T_x}$; it is clear that $v$ is coupled if $C(v) = 1$ and ghosted if $G(v) = 1$. If $C(v) = G(v) = 0$, then $v$ can be either deleted or be in the subtree of some coupled point and it is not edited by $\beta(M^*)$ since it does not appear in $\widetilde{T}_{xM^*}$. One simple way to take care of this difference is to simply considered deleted all points that are not paired nor ghosted, and then subtract the cost of the points which are not supposed to be deleted, that is, the ones in $sub(v)$ with $C(v) = 1$. In other words, the vertex $v$ is considered deleted if $D(v) = 1$, but, if $C(v) = 1$, we must correct the total cost by:

$$||sub_{T_x}(v)|| = \sum_{v' < v} d(\varphi_{T_x}(v'), 0)$$

Now we calculate the costs associated to these three possibilities. If $G(v) = 1$, $v$ is ghosted then the cost of such edit is zero. If $D(v) = 1$ the cost of this edit is $d(\varphi_{T_x}(v), 0)$. If $v$ is coupled, and so $\delta_{i,j}^{v,w} = 1$ for some unique $i, w, j$, the cost is :

$$\Delta_{i,j}^{v,w} = d(\oplus_{v' \in \zeta_v^{v_i}} \varphi_{T_x}(v'), \oplus_{w' \in \zeta_w^{w_j}} \varphi_{T_y}(w'))$$

Thus, the contribution of coupled points is $F^C(\delta)$ and the contribution of deleted points is $F^D(\delta) - F^-(\delta)$.

Now it is straightforward to write down the linear function that calculates the cost of $\beta(M^*)$: $F^\beta(\delta) := F^C(\delta) + F^D(\delta) - F^-(\delta)$. Lastly, $F^S(\delta)$ takes into account the value of $d_E(T_v, T_w)$, if $v$ and $w$ are coupled.

By Theorem 2, combined with Proposition 4, the solution of the following optimization problem:

$$\min_{\delta \in \mathcal{K}} F^S(\delta) + F^\beta(\delta) \tag{2.6}$$

is equal to $d_E(T_x, T_y)$.

■

*Proof of* Proposition 4.

Having fixed a leaf $l$, the constraint in Equation (2.2) allows for at most one path $\zeta_v^{v_{i+1}} \subset \zeta_l$ which is kept after the editing induced by the variables equal to 1. Moreover if $\delta_{i,j}^{v'',w} \in \Phi(v) \cap \Phi(v')$, then $v = v_i'' = v'$. Thus, variables are added at most one time in the constraint. Which means that for any $v' \in V_{T_x}$, we are forcing that $v'$ can be an internal vertex of at most one of the kept segments $\zeta_v^{v_{i+1}}$. In other words if two "kept" segments $\zeta_v^{v_{i+1}}$ and $\zeta_{v'}^{v'_{i'+1}}$ intersect each other, it means that they just share the upper extreme $v_{i+1} = v'_{i'+1}$. These facts together imply that (if the constraints are satisfied) the edits induced on $T_x$ by $\delta_{i,j}^{v,w} = 1$ and $\delta_{i',j'}^{v',w'} = 1$ are always compatible. Lastly, by noticing that if $\delta_{i,j}^{v,w} \in \Phi(v')$ then $\delta_{i,j'}^{v,w'} \in \Phi(v')$ for all other possible $w'$ and $j'$, we see that every segment $\zeta_v^{v_i}$ is paired with at most one segment $\zeta_w^{w_j}$, and viceversa.

As a consequence, at most one point on the path $\zeta_{v'}$ is coupled in $M^*$, for any vertex $v'$ in any of the tree structures, guaranteeing the antichain condition. Moreover any point of $T_x$ which is in $M^*$ is assigned to one and only point of $T'_y$ and viceversa. The edits induced by $\delta_{i,j}^{v,w} = 1$ clearly satisfy properties (M2)-(M4). Passing to $\widetilde{T}_{M^*}$ and $\widetilde{T'}_{M^*}$, also (M1) is satisfied.

■

*Proof of* Proposition 3.

In a full binary tree structure, at each level $l$ we have $2^l$ vertices. Let $L = len(T)$ and $L' = len(T')$. We have that, for any vertex $v \in V_T$ at level $l$, the cardinality of the path from $v$ to any of the leaves in $sub_T(v)$ is $L - l$ and the number of leaves in $sub_T(v)$ is $2^{L-l}$.

So, given $v \in V_T$ at level $l$ and $w \in V_{T'}$ at level $l'$, to calculate $d_E(sub_T(v), sub_{T'}(w))$ (having already $W_{vw}$) we need to solve a integer linear problem with $2^{L-l} \cdot (L-l) \cdot 2^{L'-l'} \cdot (L' - l')$ variables and $2^{L-l} + 2^{L'-l'}$ linear constraints.

Thus, to calculate $d_E(T, T')$, we need to solve $(2^{L+1} - 1) \cdot (2^{L'+1} - 1)$ linear integer optimization problems, each with equal or less than $2^L \cdot L \cdot 2^{L'} \cdot L'$ variables and equal or less that $2^L + 2^{L'}$ constraints. Substituting $L = log_2(N)$ and $L' = log_2(M)$ in these equations gives the result. ■

## 2.11 Merge Trees

Denote $\#C$ the cardinality of a finite set $C$. Given a fixed basis vector spaces filtration $\{(A_t, a_t)\}_{t\in\mathbb{R}}$, with maps $\psi_t^{t'} : A_t \to A_{t'}$, we build a merge tree $(T, h_T)$ which represents it up to isomorphisms. The tree structure $T$ and the height function $h_T$ are built along the following rules:

- set a leaf with height $t_0$ for every element in $a_{t_0}$;
- for every $a_k^{t_{i+1}} \in a_{t_i+1}$ such that $a_k^{t_{i+1}} \notin Im(\psi_{t_i}^{t_{i+1}})$, set a leaf with height $t_{i+1}$;
- for every $t_i$ such that $\psi_{t_i}^{t_{i+1}}(a_k^{t_i}) = \psi_{t_i}^{t_{i+1}}(a_s^{t_i})$, with $a_k^{t_i}$ and $a_s^{t_i} \in a_{t_i}$, set a vertex with height $t_{i+1}$, where the vertices associated to respectively $(\psi_{t^k}^{t_{i+1}})^{-1}(a_k^{t_i})$ and $(\psi_{t^s}^{t_{i+1}})^{-1}(a_k^{t_i})$ merge. Where $t^k = \min\{t_j | \#(\psi_{t_j}^{t_i})^{-1}(a_k^{t_i}) = 1\}$ and $t^s = \min\{t_j | \#(\psi_{t_j}^{t_i})^{-1}(a_s^{t_i}) = 1\}$.

The last merging happens at height $t_n$, which is the root of the tree structure. In this way, from $\{(A_t, a_t)\}_{t\in\mathbb{R}}$, we obtain a merge tree which is unique up merge tree isomorphism. Viceversa, through the merge tree $(T, h_T)$ we can build a fixed basis vector spaces filtration by cutting $(T, h_T)$ at every height $t$ and taking as $a_t = \{e_1, \ldots, e_k\}$ the edges in $E_T$ which are met at height $t$. The vector spaces are generated over $\mathbb{K}$ by

$a_t$. The maps are then given by the tree structure: if two edges merge at height $t$, then, at height $t$, the two corresponding basis elements are sent into the edge in which they merge. Otherwise the edges are just sent into themselves. The root $r_T$ gives the basis element at height $h_T(r_T)$.

## 2.12 Persistence Diagrams

Persistence diagrams are arguably among the most well known tools of TDA; for a detailed survey see, for instance, Edelsbrunner and Harer (2008).

A zero-dimensional persistence diagram extracted from a filtration of topological spaces $\{X_t\}_{t\in\mathbb{R}}$, with $X_t \subset X_{t'}$ if $t \leqslant t'$, represents, up to isomorphism of sequence the vector spaces filtration $\{H_0(X_t)\}_{t\in\mathbb{R}}$. Loosely speaking it is a collection of points $(c_x, c_y)$ in the first quadrant of $\mathbb{R}^2$, with $c_y > c_x$ and such that: $c_x$ is the value of $t$ corresponding to the first appearance of a path connected component in $X_t$ (birth), while $c_y$ is the "time" $t$ where the same class merges with a different class appeared before $c_x$ (death). A similar definition holds for homology classes in higher dimensions. For details about homology see Hatcher (2000). The convention that states that the "older" components survive and the "younger" die, is called *elder rule*.

CHAPTER 3

# Functional Data Representation with Merge Trees

The content of this chapter is also part of the paper Pegoraro and Secchi (2021).

In this chapter we face the problem of representation of functional data with the tools of algebraic topology. We represent functions by means of merge trees and this representation is compared with that offered by persistence diagrams. We show that these two structures, although not equivalent, are both invariant under homeomorphic re-parametrizations of the functions they represent, thus allowing for a statistical analysis which is indifferent to functional misalignment. We employ the metric for merge trees defined in Chapter 2 and we prove a few theoretical results related to its specific implementation when merge trees represent functions. To showcase the good properties of our topological approach to functional data analysis, we first go through a few examples using data generated *in silico* and employed to illustrate and compare the different representations provided by merge trees and persistence diagrams, and then we test it on the Aneurisk65 dataset replicating, from our different perspective, the supervised classification analysis which contributed to make this dataset a benchmark for methods dealing with misaligned functional data.

## 3.1 Introduction

Since the publication of the seminal books by Ramsay and Silverman (Ramsay and Silverman, 2005) and Ferraty and Vieu (Ferraty and Vieu, 2006), Functional Data Analysis (FDA) has become a staple of researchers dealing with data where each statistical unit is represented by the measurements of a real random variable observed on a fine grid of points belonging to a continuous, often one dimensional, domain $D$. In FDA these individual data are better represented as the sampled values of a function defined on $D$ and with values in $\mathbb{R}$. Hence, at the onset of any particular functional data analysis stands the three-faceted problem of *representation*, described by: (1) the smoothing of the raw and discrete individual data to obtain a functional descriptor of each unit in the data set, (2) the identification of a suitable embedding space for the sample of functional data thus obtained and, finally, (3) the eventual alignment of these functional data consistently with the structure of the embedding space. As a reference benchmark of the typical FDA pipeline applied to a real world dataset, we take the paper by Sangalli et al. (2009b) where the first functional data analysis of the AneuRisk65 dataset is illustrated.

Smoothing is the first step of a functional data analysis. For each statistical unit, individual raw data come in the form of a discrete set of observations regarded as partial observations of a function. Smoothing is the process by means of which the analyst generates the individual functional object out of the raw data. This functional object will be the atom of the subsequent analysis, a point of a functional space whose structure is apt to sustain the statistical analysis required by the problem at hand. A common approach to obtain functional representations is to fit the data with a member of a finite dimensional functional space generated by some basis, for instance, splines or trigonometric polynomials. Signal-to-noise ratio and the degree of differentiability required for the functional representation, as well as the structure of the embedding space, drive the smoothing process. Functional representations interpolating the raw data are of no practical use when the analysis requires to consider functions and their derivatives or, for instance, the natural embedding space is Sobolev's; see, for instance, Sangalli et al. (2009a) for a detailed analysis of the trade-off between goodness of fit and smoothness of the functional representation when dealing with the Aneurisk65 dataset.

Functional data express different types of variability (Vantini, 2009) which the analyst might want to decouple before carrying on the statistical analysis. Indeed the Aneurisk65 dataset is by now considered a benchmark for methods aimed at the identification of *phase* and *amplitude variation* (see the Special Section on Time Warpings and Phase Variation on the Electronic Journal of Statistics, Vol 8 (2), and references therein). In many applications phase variation captures ancillary non-informative variability which could alter the results of the analysis if not properly taken into account (Lavine and Workman, 2008; Marron et al., 2014). A common approach to this issue is to embed the functional data in an appropriate Hilbert space where equivalence classes are defined, based on a notion of *alignment* or *registration*, and then to look for the most suitable representative for any of these classes (Marron et al., 2015). Such approach

evokes ideas from shape analysis (Dryden and Mardia, 1998) and pattern theory (Ripley and Grenander, 1995), where configurations of landmark points are identified up to rigid transformations and global re-scalings. In close analogy with what has been done for curves (Michor et al., 2007; Srivastava et al., 2010), functions defined on compact real intervals $D$ are aligned by means of warping functions mapping $D$ into another interval, that is, they are identified up to some re-parametrization. Different kinds of warping functions have been investigated: affine warpings are studied for instance in Sangalli et al. (2010) while more general diffeomorphic warpings have been introduced in Srivastava et al. (2011). Once the *best* representatives are selected, the analysis is carried out on them leveraging the well behaved Hilbert structure of the embedding space. Classically, the optimal representatives are found by minimizing some loss criterion with carefully studied properties (Sangalli et al., 2014). This approach however has some limitations, arising from the fact that the metric structure of the embedding space might not be compatible with the equivalence classes collecting aligned functions (Yu et al., 2013). An alternative is to employ metrics directly defined on equivalence classes of functions such as the Fisher Rao metric, originally introduced for probability densities (Srivastava et al., 2007), which allows for the introduction of diffeomorphic warpings (Srivastava et al., 2011). It must be pointed out that all these ways of dealing with the issue of ancillary phase variability encounter some serious challenges when the domain $D$ is not a compact real interval.

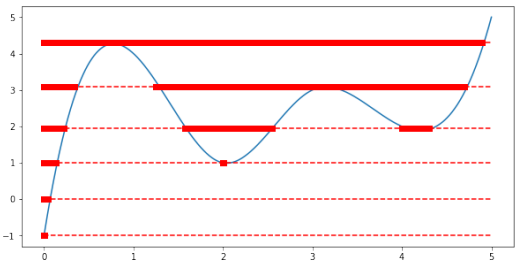
A different approach to the problem of phase variation is to capture the information content provided by a functional datum by means of a statistic which is insensitive to the functional data re-parametrization, but sufficient for the analysis. Algebraic topology can help since it provides tools for identifying information which is invariant to deformations of a given topological space (Hatcher, 2000). Topological Data Analysis (TDA) is a quite recent field in data analysis and consists of different methods and algorithms whose foundations rely on the theory developed by algebraic topology (Edelsbrunner and Harer, 2008). The main source of information collected by TDA algorithms are homology groups (see, for instance, Hatcher (2000)) with fields coefficients which, roughly speaking, count the number of holes (of different kinds) in a topological space. For instance zero dimensional holes are given by path connected components and one dimensional holes are given by classes of loops (up to continuous deformations) which cannot be shrunk to one point. One of the most interesting and effective ideas in TDA is that of *persistent homology* (Edelsbrunner et al., 2002): instead of fixing a topological space and extracting the homology groups from that space, a sequence of topological spaces is obtained along various pipelines, and the evolution of the homology groups is tracked along this sequence. The available pipelines are many, but the one which is most interesting for the purposes of this work is that concerning real valued functions. Let the domain $D$ be a topological space $X$ and consider a real valued function defined on $X$, $f : X \to \mathbb{R}$. One can associate to $f$ the sequence of topological spaces given by the sublevel sets $X_t = f^{-1}((-\infty, t])$, with $t$ ranging in $\mathbb{R}$. The evolution of the connected components along $\{X_t\}_{t\in\mathbb{R}}$ is thus analysed for the purpose of generating a topological

representation of $f$.

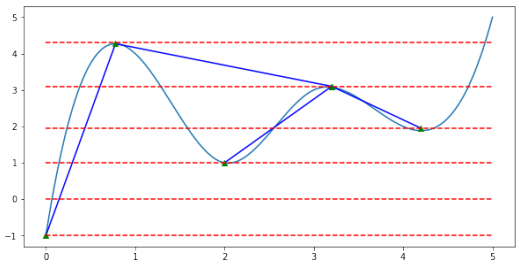
In this work, we consider specific topological representations of $f$ constructed along this general scheme and we show that they are invariant with respect to homeomorphic warpings of the domain $X$. Moreover, these representations are also able to separate big shape features of $f$ from small oscillations; the overall shape of the function captured by the topological representations we will deal with is unaffected by the presence of smaller oscillations, which are captured as well, but separately. These two properties make the TDA approach pursued in this manuscript a candidate for the representation of functional data, indeed a robust competitor able to deal in a natural way with phase variation and insensitive to the fine tuning of the preliminary smoothing phase, since functional features likely generated by overfitted representations are easily identified as ancillary in the subsequent topological representation.

To allow for the statistical analysis of functional data summarised by their topological representations, we need to embed the latter in a metric space. The choice of persistence diagrams (PD) (Cohen-Steiner et al., 2007) as summaries obtained through persistent homology drives many successful applications (Bhattacharya et al., 2015; Chung et al., 2009; Kramár et al., 2013; Pokorny et al., 2015; Wang et al., 2018; Xia et al., 2016), although other topological summaries are in fact known in the literature (Adams et al., 2017; Bubenik, 2015; Chazal et al., 2015). In this work we exploit a topological alternative – not equivalent – to a persistence diagram, called *merge tree.* Merge trees representations of functions are not new (Morozov and Weber, 2013) and are obtained as a particular case of Reeb Graphs (Biasotti et al., 2008; Shinagawa et al., 1991). Different frameworks have been proposed to work with merge trees (Beketayev et al., 2014; Morozov et al., 2013), mainly defining a suitable metric structure to compare them (Gasparovic et al., 2019; Touli, 2020). However all such metrics have a very high computational cost, causing a lack of examples and applications even when approximation algorithms are available (Touli and Wang, 2018), or they require complex workarounds to be effectively used (Sridharamurthy et al., 2020). We employ the metric for merge trees introduced in Chapter 2, showing that its computational complexity is reasonable when the trees involved are not too large. When working with representations of data, it is fundamental to study the behaviour of the operator which maps the single datum into the chosen representation to assess which kind of information is transferred from the initial data to the space of representations. For this reason we develop a new theoretical analysis on the stability/continuity of merge trees with respect to perturbations of the original functions. We also carry out examples to showcase differences between merge trees and persistence diagrams of functions. Having devoted the initial sections of this work to the understanding of the behaviour of these topological tools, we finally tackle, with our TDA approach, the benchmark functional classification case study detailed in Sangalli et al. (2009b).

The dissertation is organized as follows. In Section 3.2, we introduce the merge tree representation of a function. In Section 3.3 we briefly recall the definition of persistence diagrams in order to draw, in Section 3.4, some comparison between them and merge



**(a)** *Sublevel sets of a function*



**(b)** *A function with its associated merge tree.*

trees, before proving the invariance property which holds true for both topological representations. In Section 3.5 we present the metric structure for the space of merge trees which is used in the examples and in the final case study. In Section 3.6 we investigate the continuity properties of the operator which assigns to a function its merge tree, with respect to the aforementioned metric. After a short Section 3.7 on a visualization trick for the graphical representation of merge trees, in Section 3.8 we propose some *in silico* examples for illustrating differences and similarities between persistence diagrams and merge trees. Lastly, in Section 3.9, we tackle the functional data classification problem explored in Sangalli et al. (2009b) and we compare their results with those obtained following the TDA approach we advocate in this paper. We finally conclude the manuscript with a discussion, in Section 3.10, which points out some ideas pertaining our topological approach to functional data analysis. Section 3.12 collects the proofs of the results of the paper.

## 3.2 Merge Trees of Functions

We now define the merge tree representation of a function. Merge trees are an already established tool in topology and, to some extent, also in statistics since dendrograms can be regarded as merge trees. Nevertheless, we are going to spend a few lines to define them, in accordance with the framework defined in Chapter 2, which differs from the classical one, found for instance in Morozov and Weber (2013). Roughly speaking, the pipeline to obtain a merge tree is the following: we transform the given function into a sequence of nested subsets and then we track the topological changes along this sequence. Such information is then turned into a tree.

The details are described in the following subsections.

### 3.2.1 Sublevel Sets

Consider a function $f : X \to \mathbb{R}$, with $X$ being any topological space. We call sublevel set at height $t \in \mathbb{R}$, the set $X_t := f^{-1}((-\infty, t]) \subset X$ . The key property of the family $\{X_t\}_{t\in\mathbb{R}}$ is that such subsets are nested: if $t \leqslant t'$ then $X_t \subset X_{t'}$. Note that the sequence

$\{X_t\}_{t\in\mathbb{R}}$ is fully determined by the shape of the function $f$; see Figure 3.1(a). In fact, for $x \in X$, $f(x) = \inf_{t\in\mathbb{R}}\{t \text{ such that } x \in X_t\}$, hence no information carried by $f$ is lost by its representation $\{X_t\}_{t\in\mathbb{R}}$.

### 3.2.2 Path Connected Components

A topological space $X$ is path connected if for every couple of points $x, y \in X$ there is a continuous curve $\alpha : [0, 1] \to X$ such that $\alpha(0) = x$ and $\alpha(1) = y$. The biggest path connected subsets contained in a topological space are called path-connected components. Path connected components are the source of information we want to track along the sequence $\{X_t\}_{t\in\mathbb{R}}$.

Given $X_t \subset X$ we call $\mathbb{U}^t = \{U_i^t\}_{i\in I}$ the set of its path-connected components, which is indexed by some set $I$. We will make some very strict assumptions on such $I$ but for now we do not need them. The main fact about path connected components is that if $X_t \subset X_{t'}$, then, for every $i$, there is a unique $j$ such that $U_i^t \subset U_j^{t'}$.

Thus, we can define:

$$\alpha_t^{t'} : \mathbb{U}^t \to \mathbb{U}^{t'}$$

such that

$$U_i^t \subset \alpha_t^{t'}(U_i^t)$$

for all $U_i^t \in \mathbb{U}^t$.

A $t \in \mathbb{R}$ is called *critical value* if, for every $\varepsilon > 0$, $\alpha_{t-\varepsilon}^t$ is not bijective.

### 3.2.3 Tree Structures

Coherently with Chapter 2, we now define what we mean with *tree* and with *merge tree*.

**Definition 15.** *A tree structure $T$ is given by a set of vertices $V_T$ and a set of edges $E_T \subset V_T \times V_T$ which form a connected rooted acyclic graph. We indicate the root of the tree with $r_T$. We say that $T$ is finite if $V_T$ is finite. The order of a vertex of $T$ is the number of edges which have that vertex as one of the extremes. Any vertex with an edge connecting it to the root is its child and the root is its father: this is the first step of a recursion which defines the father and children relationship for all vertices in $V_T$. The vertices with no children are called leaves or taxa. The relation $father > child$ induces a partial order on $V_T$. The edges in $E_T$ are identified in the form of ordered couples $(a, b)$ with $a < b$. A subtree of a vertex $v$ is the tree structure whose set of vertices is $\{x \in V_T | x \leqslant v\}$.*

**Definition 16.** *A finite tree structure T coupled with a monotone increasing height function $h_T : V_T \to \mathbb{R}$ is called merge tree.*

Let us see how, starting from a real valued function $f : X \to \mathbb{R}$, we can represent it by means of a merge tree. We use the following notation: given a finite set $C$, then $\#C$ is its cardinality.

Consider $f : X \to \mathbb{R}$, and assume $X$ to be a path connected topological space and $f$ a *tame* function. We recall that a function is tame if for every $X_t$, the set $\mathbb{U}^t$ is finite and along the sequence $\{X_t\}_{t\in\mathbb{R}}$ there are only a finite set of critical values. The idea is that, since path-connected components in every $X_t$ can only arise, merge with others, or stay the same, it is quite natural to represent this merging structure with a tree structure. However, a tree structure $T$ is not enough to represent the information contained in $\mathbb{U}^t$ and $\alpha_t^{t'}$, so we also define a monotone increasing height function $h_T : V_T \to \mathbb{R}$ which encodes the critical values $t_0 < ... < t_n$ of $f$.

The tree structure $T$ and the height function $h_T$ are built along the following rules:

- set a leaf with height $t_0$ for every element in $\mathbb{U}^{t_0}$;
- for every $U \in \mathbb{U}^{t_i+1}$ such that $U \notin Im(\alpha_{t_i}^{t_{i+1}})$, set a leaf with height $t_{i+1}$;
- for every $t_i$ such that $\alpha_{t_i}^{t_{i+1}}(U) = \alpha_{t_i}^{t_{i+1}}(U')$, with $U$ and $U'$ in $\mathbb{U}^{t_i}$, set a vertex with height $t_{i+1}$, where vertices associated to $(\alpha_{t^U}^{t_i})^{-1}(U)$ and $(\alpha_{t^{U'}}^{t_i})^{-1}(U')$ merge. With $t^U = \min\{t_j \big| \#(\alpha_{t_j}^{t_i})^{-1}(U) = 1\}$ and $t^{U'} = \min\{t_j \big| \#(\alpha_{t_j}^{t_i})^{-1}(U') = 1\}$.

The last merging happens at height $t_n$ and, since $X$ is path connected, at height $t_n$ there is only one point, which is the root of the tree structure.

Look at Figure 3.1(b) for a first example. The height function is given by the dotted red lines. We can appreciate that the merge tree of $f$ is heavily dependent on the shape of $f$, in particular on the displacement of its maxima and minima.

### 3.2.4 Isomorphism classes

Before continuing we must decide on the topological information which we regard as equivalent. In other words, which merge trees we want to distinguish and which we do not. This step is essential and decisive to tackle the phase variation problem presented in the introduction: to select information that is insensitive to some kind of transformation amounts to defining classes of functions which are represented by the same tree. We opt for a very general solution: we remove from the vertices of the tree any information regarding the connected components they are associated to, for instance, size, shape, position, the actual points contained etc..

**Definition 17.** *Two tree structures $T$ and $T'$ are isomorphic if there exists a bijection $\eta : V_T \to V_{T'}$ inducing a bijection between the edges sets $E_T$ and $E_{T'}$: $(a, b) \mapsto (\eta(a), \eta(b))$. Such $\eta$ is an isomorphism of tree structures.*

**Definition 18.** *Two merge trees $(T, h_T)$ and $(T', h_{T'})$ are isomorphic if $T$ and $T'$ are isomorphic as tree structures and the isomorphism $\eta : V_T \to V_{T'}$ is such that $h_T = h_{T'} \circ \eta$. Such $\eta$ is an isomorphism of merge trees.*

The rationale behind Definition 17, and the equivalence classes of isomorphic merge trees it generates, is analogous to that moving the introduction of persistence diagrams,

where no specific information about individual path connected components is retained (see Section 3.3 for more details). Moreover, Definition 17 does not require any additional structure for the space $X$. Other choices are possible; for instance, if $X = \mathbb{R}$ the path connected components of $f$ could be given a natural ordering.

### 3.2.5 Height and Weight Functions

A final step is needed to complete the construction of the merge trees we are going to use in the following sections. The height function $h_T$ of a tree $T$ takes values in $\mathbb{R}$, but this is not an *editable* space, as defined in Chapter 2. Hence we transform the height function $h_T$ into a weight function $w_T$ defined on $V_T$ and such that the image of $V_T - \{r_T\}$ is a subset of the editable space $\mathbb{R}_{\geqslant 0}$.

For every vertex $v \in V_T - \{r_T\}$, we consider the unique edge between $v$ and its father $w$ and we define $w_T(v) = h_T(w) - h_T(v)$. We set $w_T(r_T) = h_T(r_T)$. Note that there is a one-to-one correspondence between $h_T$ and $w_T$. Finally, the monotonicity of $h_T$ guarantees that $w_T(v) \in \mathbb{R}_{\geqslant 0}$, for all $v \in V_T - \{r_T\}$.

The height function introduced in Definition 16 turns out to be quite natural for the definition of a merge tree, but from now on we replace the height function $h_T$ with the induced weight functions $w_T$.

## 3.3 Persistence Diagrams

Persistence diagrams are arguably among the most well known tools of TDA; for a detailed survey see, for instance, (Edelsbrunner and Harer, 2008). We here briefly introduce them since in the following sections we use them to draw comparisons with merge trees.

Loosely speaking a persistence diagram is a collection of points $(c_x, c_y)$ in the first quadrant of $\mathbb{R}^2$, with $c_y > c_x$ and such that: $c_x$ is the $t$ corresponding to the first appearance of an homology class in $X_t$ (birth), while $c_y$ is the $t$ where the same class merges with a different class appeared before $c_x$ (death). Homology classes are a generalization of path-connected components to "holes in higher dimension"; path-connected components can be seen as zero dimensional holes. For more details see Hatcher (2000).

In this work we focus on persistence diagrams associated to path-connected components, since we want to compare them with the merge trees introduced in the previous section. Given a function $f : X \to \mathbb{R}$, we associate to $f$ the zero dimensional persistence diagram ($PD(f)$) of the sequence of sublevel sets $\{X_t\}_{t\in\mathbb{R}}$. In such representation there is no information about which path-connected component merges with which; in fact a component represented by the point $(c_x, c_y)$, at height $c_y$ could merge with any of the earlier born and still alive components. Of course this collection of points depends on the shape of the function and in particular depends on its amplitude and the number of its oscillations. See Figure 3.2. Note that, while for merge trees one needs to

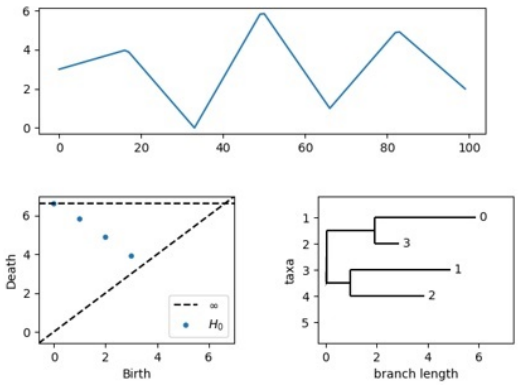



**Figure 3.2:** *A function with its associated persistence diagram (left) and merge tree (right). On the PD axes we see the birth and death coordinates of its points. The plot of the merge tree features the length of its branches (given by the weight function) on the horizontal axis, and the leaves (taxa) are displaced on the vertical axis. The vertical axis scale is only for visualization purposes.*

be careful and consider appropriate isomorphism classes so that the representation does not depend on, for instance, the names chosen for the vertices (that is, the set $V_T$), this issue does not appear with persistence diagrams. Topological features are represented as points in the plane, without labels or other kinds of set-dependent information. Thus, two persistence diagrams are isomorphic if and only if they are made of the same set of points.

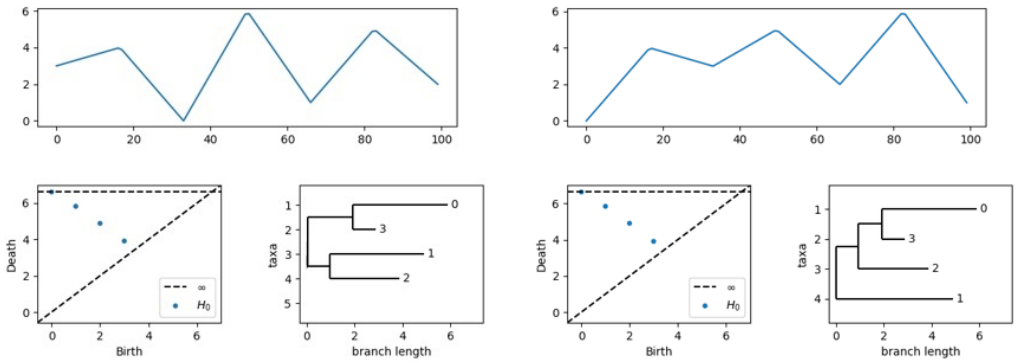
## 3.4 Properties

In this section we state the main invariance result anticipated in the introduction and we also point out a few differences between persistence diagrams and merge trees.

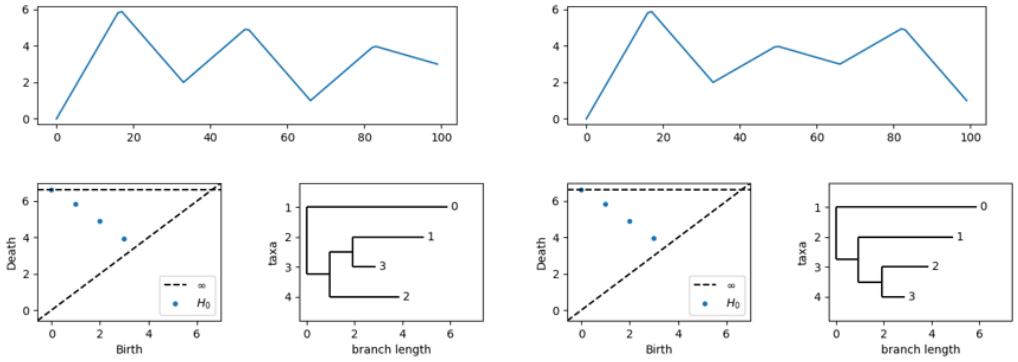
**Proposition 5** (Invariance)**.** *The (isomorphism class of the) merge tree and the persistence diagram of the function $f : X \to \mathbb{R}$, are both invariant under homeomorphic re-parametrization of $f$.*

**Remark 13.** *As an immediate consequence of Proposition 5 we obtain that, if the functions $f$ and $g$ can be aligned by means of an homeomorphism, that is if $f = g \circ \eta$ being $\eta$ an homeomorphism, then their associated merge trees $T_f$ and $T_g$ are isomorphic and the same holds for $PD(f)$ and $PD(g)$.*

In other words, we can warp, deform, move the domain $X$ of a function $f$ by means of any homeomorphism, and this will have no effect on its associated PD or merge tree.




**(a)** $f_0$ **(b)** $f_1$




**(c)** $f_2$ **(d)** $f_3$

**Figure 3.3:** *We compare four functions; they are all associated to the same $PD$ but to different merge trees. Functions are displayed in the first row of each subplot, while on the second we have on the left the associated $PD$ and on the right the merge tree.*

As a consequence, if each element of a sample of functions is represented by its merge tree, or by its persistence diagram, one can carry out the statistical analysis without worrying about possible misalignement, that is without first singling out, for each function of the sample, the specific warping function, identified by an homeomorphism, which decouples its phase and amplitude variabilities.

Despite sharing this important invariance property, a persistence diagram and a merge tree are not equivalent representations of a function. Indeed, persistence diagrams do not record information about the merging components. This is an important difference, since merge trees can capture also this local structure of a function (see Figure 3.3). Moreover, the next proposition proves that the information contained in the persistence diagram of a function $f$ can be retrieved from the merge tree associated to $f$, but the converse is not true as shown in Figure 3.3

**Proposition 6.** *For all $f : X \to \mathbb{R}$, the associated $PD(f)$ can be obtained by the*

*associated merge tree* $T_f$.

Thus, if two functions induce isomorphic merge trees, they also have the same persistence diagrams.

## 3.5 Metrics

We want to analyze sets of functions using merge trees and PDs, exploiting metrics which have already been defined respectively in Chapter 2 and in Cohen-Steiner et al. (2010). Here we quickly present such metrics, with a special focus on the metric for merge trees, since we use it to develop novel stability results in the next sections.

### 3.5.1 Metrics for Persistence Diagrams

The space of persistence diagrams can be given a metric structure by means of a family of metrics which derives from Wasserstein distances for bivariate distributions.

Given two diagrams $D_1$ and $D_2$, the expression of such metrics is the following:

$$W_p^p(D_1, D_2) = inf_\gamma \sum ||x - \gamma(x)||_\infty^p$$

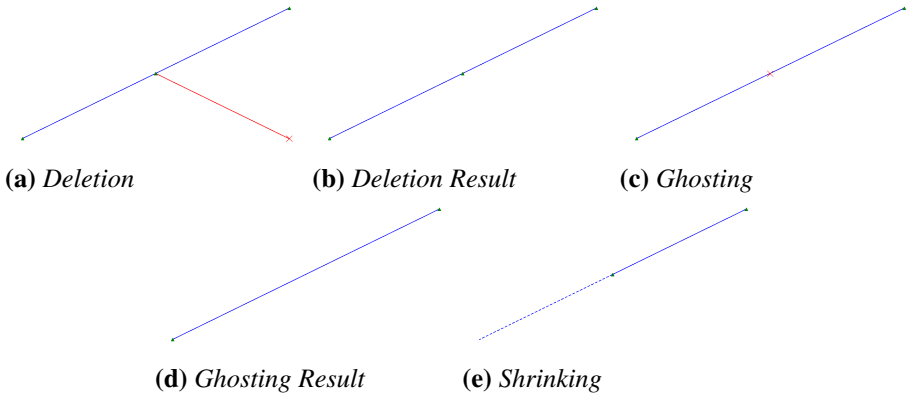
where $\gamma$ are functions partially matching points between diagrams $D_1$ and $D_2$, and matching remaining points with the line $y = x$ on the plane. In other words we measure the distances between the points of the two diagrams, pairing each point of a diagram either with a point on the other diagram, or with a point on $y = x$. Each point can be matched once and only once. The minimal cost of such matching provides the distance.

### 3.5.2 Metric for Merge Trees

The metric for tree-like objects defined in Chapter 2 is based on edit distances (Bille, 2005; Hong et al., 2017): they allow for modifications of a starting object, each with its own cost, to obtain a second object. Merge trees equipped with their weight function $w_T$, as defined in Section 3.2.5, fit into this framework; hence the space of merge trees can be endowed with a metric based on an edit distance and called $d_E$ in the following.

The distance $d_E$ is very different from previously defined edit distances, since it is specifically designed for comparing topological summaries. It satisfies the property of *topological stability* (see Section 3.5.2), roughly meaning that all points which are topologically irrelevant can be eliminated by a merge tree without paying any cost. To make things more formal we here introduce the edits, as defined in Chapter 2, but directly in the context of merge trees, since they were originally defined for more general objects.

The edits are the followings:



**(a)** *Deletion* **(b)** *Deletion Result* **(c)** *Ghosting*

**(d)** *Ghosting Result* **(e)** *Shrinking*

**Figure 3.4:** *(a)→(e) form an edit path made by one deletion , one ghosting and a final shrinking.*

- *shrinking* an edge means changing the weight value of the edge with a new positive value. The inverse of this transformation is the shrinking which restores the original edge weight.

- *deleting* an edge $(v_1, v_2)$ results into a new tree, with the same vertices apart from $v_1$ (the lower one), and with the father of the deleted vertex which gains all of its children. This edit cannot be done on the root. With a slight abuse of language, we might also refer to this edit as the deletion of the vertex $v_1$, which indeed means deleting the edge between $v_1$ and its father.

  The inverse of deletion is the *insertion* of an edge along with its child vertex. We can insert an edge at a vertex $v$ specifying the child of $v$ and its children (that can be either none or any portion of the children of $v$) and the weight of the edge.

- Lastly, we can eliminate an order two vertex $v$, that is a father with an only child, connecting the two adjacent edges which arrive and depart from $v$. The weight of the resulting edge is the sum of the weights of the joined edges. This transformation is the *ghosting* of the vertex $v$ and it cannot be done on the root. Its inverse transformation is called the *splitting* of an edge.

**Remark 14.** *Edit operations are not globally defined as operators mapping merge trees into merge trees. They are defined on the individual tree. Similarly, their inverse is not the inverse in the sense of operators, but it indicates that any time we travel from a tree $T$ to a tree $T'$ by making a sequence of edits, we can also travel the inverse path going from $T'$ to $T$ and restore the original tree.*

The costs of the edit operations are defined as follows:

- the cost of shrinking an edge is equal to the absolute value of the difference of the two weights;

- for any deletion/insertion, the cost is equal to the weight of the edge deleted/inserted;
- the cost of ghosting is zero.

The root of a merge tree can only be edited by changing its weight and the cost of such editing is the absolute value of the weight change.

Given a tree $T$ we can edit it, thus obtaining another tree, on which we can apply a new edit to obtain a third tree and so on. Any finite composition of edits is called *edit path*. The cost of an edit path is the sum of the costs of its edit operations. Putting all the pieces together, we can define the edit distance $d_E$ as:

$$d_E(T, T') = inf_{\gamma \in \Gamma(T,T')} cost(\gamma)$$

where $\Gamma(T, T')$ indicates the set of edit paths which start in $T$ and end in $T'$.

#### Order Two Vertices

The null cost of ghosting guarantees that order $2$ vertices are completely irrelevant when computing the cost of an edit path.

**Definition 19.** *If there is an edit path from the tree $T$ to the tree $T'$ consisting only of ghosting edits, we say that the two trees are equal up to order $2$ vertices. By definition, the length of the edit path starting in $T$ and ending in $T'$ is equal to $0$.*

In Chapter 2 it is proved that $d_E$ is a metric on the space of merge trees, identified up to order $2$ vertices. As explained in Chapter 2, the fact that order $2$ vertices are irrelevant is precisely what makes the metric $d_E$ suitable for comparing merge trees and is fundamental to obtain the results of the following Section 3.6.

## 3.6 Pruning & Stability

As stated in the introduction of the paper, any time we use a data representation – or we further transform a representation – it is important to understand and explore the properties of the operators involved. In particular, in this section we establish some continuity properties for the operator $f \mapsto T_f$, which maps a function to its merge tree. Conditional on the topology endowing the functional space where the function $f$ is embedded, these properties dictate how the variability between functions is captured by the variability between their merge tree representations.

Proposition 5 implies that the merge tree representation of a function $f$ is unaffected by a large class of warpings of its domain, which would strongly perturb $f$ if it was embedded, for instance, in an $L_p$ space, with $p \neq \infty$. As an example, if $f : \mathbb{R} \to \mathbb{R}$ has compact support, shrinking $f$ by setting $f_n(x) = f(x \cdot \lambda_n)$ with $\lambda_n \to +\infty$, produces

no effect on the merge tree representation of $f$ since $T_{f_n} = T_f$, while the $p$-norm of $f_n$ goes to zero.

It might therefore be more natural to study the behavior of $f \mapsto T_f$ endowing the space of functions $f : X \to \mathbb{R}$ with the topology of pointwise convergence, which captures pointwise closeness between functions. This topology, available for any domain $X$, has also the advantage of showing the effect of pointwise noise on merge tree representations.

### 3.6.1 Pruning

We know that, given $f$, the merge tree $T_f$ will mostly depend on the critical points of $f$: as the number of spikes of $f$ grows, also the size of the tree grows, while the weights of its branches grow with the height of the spikes. Similarly, if two functions $f, g : X \to \mathbb{R}$ are pointwise $\varepsilon$ close, we can say that the shape of the functions is the same up to spikes of height $2 \cdot \varepsilon$. Each such spike would cause the birth of a leaf whose branch is shorter than $2 \cdot \varepsilon$; the trees must therefore be the same up to branches of weight $2 \cdot \varepsilon$. These considerations move the idea of pruning, which consists of removing unessential edges from a tree.

Given a merge tree without order 2 vertices, we delete the small weight leaves, that is those whose weight is smaller than or equal to a given fixed threshold. However, if two or more small weight leaves are siblings, we only remove that of smallest weight, or one of the leaves chosen at random if they have the same weight, and then ghost its father if it becomes an order 2 vertex. This procedure is done recursively until no small weight leaves are found. Note that removing only one leaf in case of siblings of small weight, prevents the possible removal of information relative to spikes of $f$ with amplitude larger than the threshold.

We can thus define the pruning operator:

$$P_\varepsilon : \mathcal{T} \to \mathcal{T}$$

such that $P_\varepsilon(T)$ is the tree obtained by pruning with threshold $\varepsilon$. Notice that $P_\varepsilon$ is idempotent, that is $P_\varepsilon(P_\varepsilon(T)) = P_\varepsilon(T)$.

**Remark 15.** *$P_\varepsilon$ is not a continuous operator. Consider $T$ formed by just one edge with weight $\varepsilon$; take $\delta > 0$ and consider $T'$, with the same tree topology as $T$ but made by one edge of weight $\varepsilon + \delta$. Now, $d_E(T, T') = \delta$ and $d_E(P_\varepsilon(T), P_\varepsilon(T')) = \varepsilon + \delta$. If we let $\delta \to 0$, then $d_E(P_\varepsilon(T), P_\varepsilon(T')) > \varepsilon$.*

For what has been said up to now, the operator $P_\varepsilon$ can be considered as a smoothing operator. We fix some threshold which we think captures meaningful shape changes in a function and then, consistently, we remove what is deemed to be noise from the representation, obtaining a more regular merge tree. This has also the effect of greatly decreasing the number of leaves of the tree, a fact that is important from the computational perspective.

### 3.6.2 Stability

Now we study the case of two merge trees $T_f$ and $T_g$ representing functions $f$ and $g$ which are pointwise $\varepsilon$ close.

The main theorem of this section is the following.

**Theorem 3.** *Let $f, g$ be tame functions defined on a path connected topological space $X$ and such that*

$$sup_{x \in X} |f(x) - g(x)| \leqslant \varepsilon.$$

*Let $T_f$ and $T_g$ be the merge trees associated to $f$ and $g$ respectively and let $N$ and $M$ be the cardinalities of $V_{T_f}$ and $V_{T_g}$.*

*Then, there exists an edit path $e_1 \circ \ldots \circ e_{N \cdot M} \in \Gamma(T_f, T_g)$ such that $cost(e_i) < 2 \cdot \varepsilon$, for $i = 1, ..., N \cdot M$.*

Theorem 3 states that if two functions are pointwise close, then we can turn the merge tree associated to the first function into the merge tree associated to the second function, using edits of small cost, at most one per vertex. Note, however, that if the two functions have a very high number of oscillations, the distance between their merge trees could still be large. Indeed if $||f_n - f||_\infty \xrightarrow{n} 0$ with $\#V_{T_{f_n}} \xrightarrow{n} \infty$, we are not guaranteed that $d_E(T_f, T_{f_n}) \to 0$. Theorem 3 however implies that, if the cardinalities $|V_{T_{f_n}}|$ are bounded, then $d_E(T_f, T_{f_n})$ indeed converges to $0$.

Problems could then arise when we expect a possibly unbound number of informative spikes, that is spikes which should not be removed by pruning. In this case, however, the computational cost of the metric $d_E$ would also be prohibitive due to the high number of leaves in the trees; indeed this supports the claim that the only practical limitation to the use of the metric $d_E$ is given by its computational cost.

### 3.6.3 Spline Spaces

We here emphasize for spline spaces the consequences of the results of the previous two subsections, since splines are often used in FDA applications for smoothing the discrete raw data profiling each statistical unit in the sample.

As already noted in the introduction, spline spaces are a preferred tool for smoothing functional data since they provide finite dimensional vector spaces of functions with convenient properties. In particular, spline functions are piecewise polynomials determined by a grid of knots; fixing the knots determines a finite upper bound for the number of critical points of the spline. Consider for instance $\mathcal{S}_n^3$, the space of piecewise cubic polynomials over a grid on $[0, 1]$ with $n$ equispaced knots. On each interval the first derivative of the function is a quadratic polynomial and thus its zero set is composed by at most two points. This means that the number of critical points of $f \in \mathcal{S}_n^3$ is at most $2(n-1)$; therefore the number of leaves of the tree $T_f$ associated to $f$ cannot be greater than $2(n-1)$. The following Corollary of Theorem 3 is in fact easily obtained:

**Corollary 3.** *Let $\mathcal{S}$ be a space of piecewise polynomial functions of some fixed degree, all defined by means of the same finite grid of nodes. Then the operator $f \in \mathcal{S} \mapsto T_f$ is continuous.*

Smoothing raw data with splines entails a delicate trade-off between being flexible, to capture the salient features of the function the raw data have been sampled from, and avoiding the introduction of artifacts, due, for instance, to noise overfitting or caused by forcing the spline to fit an abrupt spike. Representing the smoothed spline function by means of a merge tree can help in handling this trade-off, by allowing the analyst a certain degree of casualness in the smoothing phase, since the small artifacts generated by a possible overfitting will then be controlled by pruning the tree.

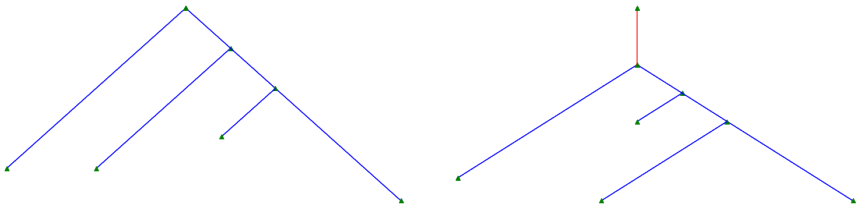
For instance, consider the problem of approximating $f : [0, 1] \to \mathbb{R}$ with a cubic spline function defined by an equispaced grid of knots. Suppose $f$ satisfies some regularity conditions, usually implied by its embedding in a Sobolev space. The parameter which controls the bias-variance trade-off is just the number of knots $n$. Many results are known in the literature concerning the uniform convergence of spline functions as the step of the grid of knots goes to zero (see for instance De Boor and Daniel (1974a)) and most of them are given in terms of a factor $1/n^\alpha$ and the norm of the derivatives or the modulus of continuity of $f$. In other words, the pointwise error can be reduced as needed by increasing $n$. When $f$ is approximated by the spline function $s_f$ with an error of $\varepsilon$ in terms of uniform norm, this means that whatever happens in intervals of $\pm\varepsilon$ around $f$ is inessential. Stated in different terms, oscillations of $s_f$ taking place in such zone are to be considered uninformative with respect to the analysis. Thus a sensible choice is to represent the function $f$ fitted by the spline $s_f$ by means of the pruned merge tree $P_{2\varepsilon}(T_{s_f})$. If $\varepsilon$ is small enough with respect to the oscillations of $f$, Theorem 3 implies that pruning $T_{s_f}$ by $2\varepsilon$ removes only inessential edges of $s_f$, without loosing important information about $f$.

The same argument applies when smoothing observations sampled from a function $f$. The analyst may allow the spline to overfit the data and then decide that oscillations under a certain amplitude are irrelevant, controlling them by pruning the merge tree associated to the fitted spline.

## 3.7 Visualization trick

Before showcasing the examples and a case study, we point out a *visualization trick* employed when we graphically represent merge trees for visual comparison and evaluation.

Given a set of functions, each represented by a merge tree, we let $M$ be the maximum value attained by any function in the set, and to each representing tree, say $T$, we add an edge connecting its root $r_T$ to a new point, which becomes the new root, at height $M$. The new edge of course is given weight $M - w_T(r_T)$. In this way all merge trees in



**(a)** *Tree structure $T$. Suppose $h_T(r_T) = 1$ and $M = 2$.*

**(b)** *Tree structure $T$ with the added edge of lenght $1$ and the new root.*

**Figure 3.5:** *On the left we can partially see a merge tree: namely we see its tree structure with the weights represented by the length of the edges. The information about the height value of the root is not visualized. On the right we see the same merge tree represented with the visualization trick: the red edge allows a visual comparison between different merge trees represented in such way.*

the dataset are "hanging" from height $M$, and can therefore be visually compared using existing libraries for trees representation. See Figure 3.5.

Moreover, apply this visualization trick to two trees $T_f$ and $T_g$. The cost of shrinking the edge of weight $M - w_{T_f}(r_{T_f})$ added to the tree $T_f$ to the corresponding edge added to the tree $T_g$ is exactly $|w_{T_f}(r_{T_f}) - w_{T_g}(r_{T_g})|$ and this is the cost of editing the roots $r_{T_f}$ and $r_{T_g}$ to match heights. Hence, the visual comparison of the merge tree representations is consistent with the metric $d_E$.

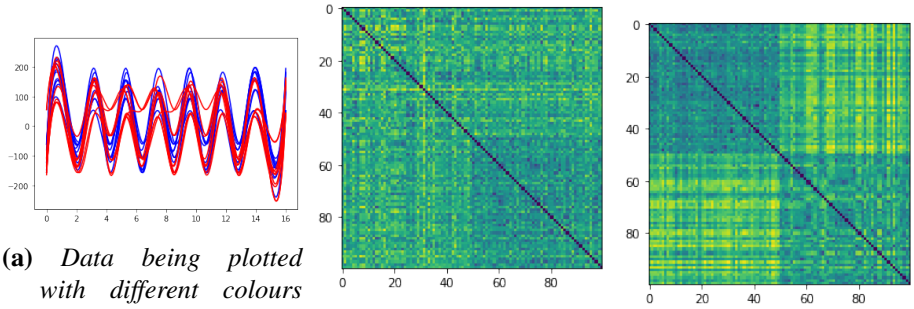
## 3.8 Examples

These examples are intended to show the differences between persistence diagrams and merge trees, already highlighted in Section 3.4.

### 3.8.1 Example I

We generate two clusters of functional data such that the membership of a function to one cluster or the other should depend on the amplitude of its oscillations and not on the merging structure of its path connected components. We then look at the matrices of pairwise distances between functions, comparing merge tree and persistence diagram representations in terms of their goodness in identifying the clustering structure.

To generate each cluster of functions, we draw, for each cluster, an independent sample of 16 critical points, 8 maxima and 8 minima, from two univariate Gaussian distributions with means equal to $+100$ for maxima and to $-100$ for minima, respectively. The standard deviations of the two Guassian distributions are the same and they are set equal to $50$. To generate a function inside a cluster, we draw a random permutation of 8 elements and according to that permutation we reorder both the set of maxima and



**(a)** *Data being plotted with different colours depending on the cluster.*

**(b)** *Merge trees pairwise distances.*

**(c)** $PDs$ *paiwise distances.*

**Figure 3.6:** *Example I. In the first row we can see few data from the two clusters. In the second row we see the matrices of pairwise distance extracted with trees and $PDs$. The data are ordered according to their cluster. It is clear how $PDs$ perform much better in separating the two clusters.*

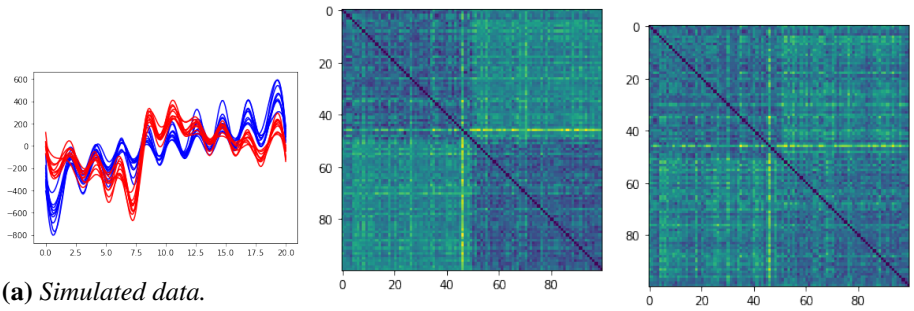
the set of minima associated to the cluster. Then, we take a regular grid of 16 nodes on the abscissa axis: on the ordinate axis we associate to the first point on the grid the first minimun, to the second the first maximum, to the third the second minimum and so on. To obtain a function we interpolate such points with a cubic spline. We thus generate $50$ functions in each cluster. The key point is that, within the same cluster, the critical points are the same but for their order, while the two clusters correspond to two different sets of critical points.

In this example, we expect that the clustering structure carried by the amplitude of the functions will be shadowed by the differences in the merging order, when adopting the merge tree representation; while persistence diagrams should perform much better because they are less sensitive to peak reordering. This is in fact confirmed by inspecting the distance matrices in Figure 3.6(b) and Figure 3.6(c).

### 3.8.2 Example II

Here we reverse the state of affairs and we set the feature for discriminating between clusters to be the merging structure of the functions. Hence, we generate two clusters of functions: the members of each cluster have the same merging structure which is however different between clusters.

To generate the two clusters of 50 functions each, we first draw an independent sample of $10$ critical values, $10$ maxima and $10$ minima, shared between the clusters. Such samples are drawn from Gaussian distributions with means $100$ and $-100$ respectively and standard deviation $200$. Given a regular grid of $20$ nodes on the abscissa axis, on the ordinate axis we associate to the first point of the grid a maximum, to the second a minimum, and so on, as is Example I. To generate every member of one cluster or the other, we add to the ordinate of each maximum or minimum critical point a random



**(a)** *Simulated data.*

**(b)** *Merge trees pairwise distances.*

**(c)** *PD pairwise distances.*

**Figure 3.7:** *Example II. In the first row we can see a few data from the two clusters. In the second row we see the matrices of pairwise distances between merge tree representations and $PDs$, respectively. The data are ordered according to their cluster. It is clear how in this example merge trees are more suitable to separate the two clusters.*

noise generated by a Gaussian with mean 0 and standard deviation $100$. Then we re-order such points following a cluster-specific order. And, lastly, we interpolate with a cubic spline. We remark that the ordering of the maxima and that of the minima now becomes essential. For the two clusters, these orderings are fixed but different and they are set as follows ($0$ indicates the smallest value and $9$ being the largest value):

- first cluster: maxima are ordered along the sequence $(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)$, minima along the sequence $(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)$;
- second cluster: maxima are ordered along the sequence$(3, 2, 1, 0, 8, 9, 7, 6, 4, 5)$, minima along the sequence $(3, 2, 1, 0, 8, 9, 7, 6, 4, 5)$.

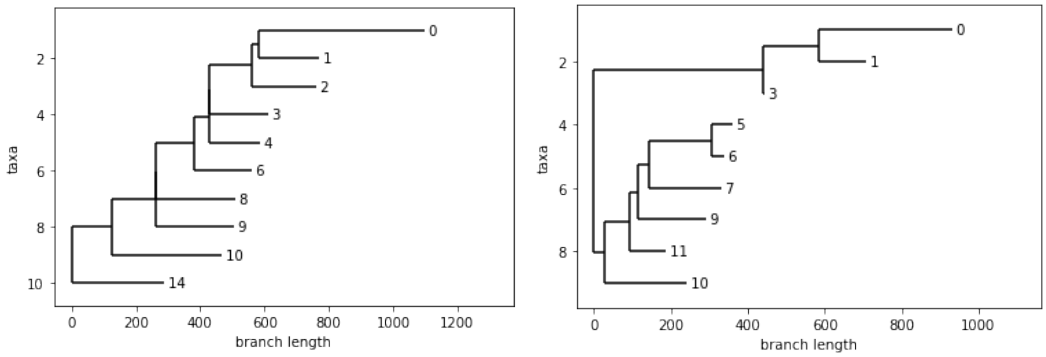
Such different orderings provide non-isomorphic tree structures for the merge trees associated to the functions of the two clusters, as we can see in Figure 3.8, while keeping a similar structure in terms of persistence diagrams.

In this example, we expect $PDs$ to be unable to recognise the clustering structure among the data; indeed, the only discriminant feature available to $PDs$ is the different height of critical points, but this bears little information about the clusters.

We can visually observe this by comparing Figure 3.7(b) with Figure 3.7(c).

## 3.9 Case Study

We now run a comparative analysis of the real world Aneurisk65 dataset. This dataset – and the clinical problem for which it was generated and studied – was first described in Sangalli et al. (2009b), but it has since become a benchmark for the assessment of FDA methods aimed at the supervised or unsupervised classification of misaligned functional




**(a)** *Tree structure of the first cluster.*

**(b)** *Tree structure of the second cluster.*

**Figure 3.8:** *Example II. The tree structures of the two clusters.*

data (see, for instance, the special issue of the Electronic Journal of Statistics dedicated to phase and amplitude variability - year 2014, volume 8). We then repeat the classification exercise illustrated in Sangalli et al. (2009b) with the double scope of comparing merge trees and persistent diagrams when used as representations of the Aneurisk65 misaligned functional data, and of evaluating the performance of these representations for classification purposes when compared with the results obtained with the more traditional FDA approach followed by Sangalli et al. (2009b).

### 3.9.1 Dataset

The data of the Aneurisk65 dataset were generated by the Aneurisk Project, a multidisciplinary research aimed at investigating the role of vessel morphology, blood fluid dynamics, and biomechanical properties of the vascular wall, on the pathogenesis of cerebral aneurysms. The project gathered together researchers of different scientific fields, ranging from neurosurgery and neuroradiology to statistics, numerical analysis and bio-engineering. For a detailed description of the project scope and aims as well as the results it obtained see its web page (`https://statistics.mox.polimi.it/aneurisk`) and the list of publications cited therein.

Since the main aim of the project was to discover and study possible relationships between the morphology of the inner carotid artery (ICA) and the presence and location of cerebral aneurysms, a set of three-dimensional angiographic images was taken as part of an observational study involving 65 patients suspected of being affected by cerebral aneurysms and selected by the neuroradiologist of Ospedale Niguarda, Milano. These 3D images where then processed to produce 3D geometrical reconstructions of the inner carotid arteries for the 65 patients. In particular, these image reconstructions allowed to extract, for the observed ICA of each patient, its centerline "raw" curve, defined as the curve connecting the centres of the maximal spheres inscribed in the vessel, along with the values of the radius of such spheres. A detailed description of the pipeline followed

to identify the vessel geometries expressed by the AneuRisk65 functional data can be found in Sangalli et al. (2014).

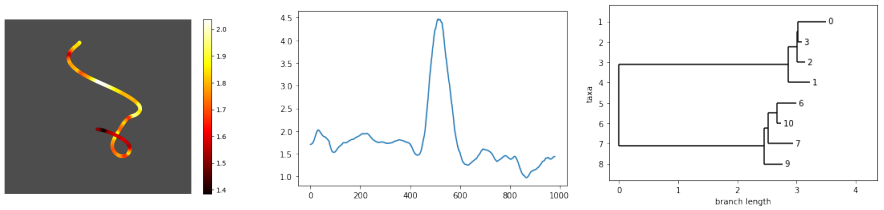
Different difficulties arise when dealing with this data. First, as detailed in Sangalli et al. (2009a), to properly capture information affecting the local hemodynamics of the vessels, the curvature of the centerline must be obtained in a sensible way. Retrieving the salient features of the centerline and of its derivatives is a delicate operation, which is heavily affected by measurement errors and reconstruction errors, due to the complex pipeline involved. Consequently the "raw" curves appear to be very wiggly and it is not obvious how to produce reasonable smooth representations. At the same time the 3D volume captured by the angiography varies from patient to patient. This is due to many factors, such as: the position of the head with respect to the instrument, which in turns depends on the suspected position of the aneurysm, the disposition of the vessels inside the head of the patient, the size of the patient. We can recognize these differences even by visual inspection in Figure 3.9: for instance, in Figure 3.9(a) and Figure 3.9(g) we see a longer portion of the ICA than in Figure 3.9(d). Therefore the reconstructed ICAs cannot be directly compared: we need methods that take into account that the centerlines are not embedded in $\mathbb{R}^3$ in the same way, and that we cannot expect potentially interesting features to appear in exactly the same spots along the centerline. This is the typical situation where one should resort to alignment.

Hence, this dataset is paradigmatic of the three-faceted representation problem highlighted in the introduction; data smoothing, embedding, and alignment present difficult challenges, which propelled a number of original works in FDA.

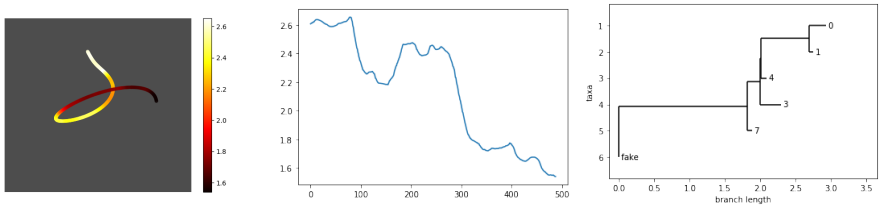
The AneuRisk65 data have been already partially processed; in particular centerlines have been smoothed following the free-knot regression spline procedure described in Sangalli et al. (2009a), and their curvatures were thus obtained after computing the first two derivatives of the smoothed curves. The data relative to the radius of the blood vessel, instead, although measured on a very fine grid of points along the centerline, is still in its raw format. Hence the AneuRisk65 data also allow us to compare the behaviour of tree representations on smoothed data and on raw data.

### 3.9.2 Analysis

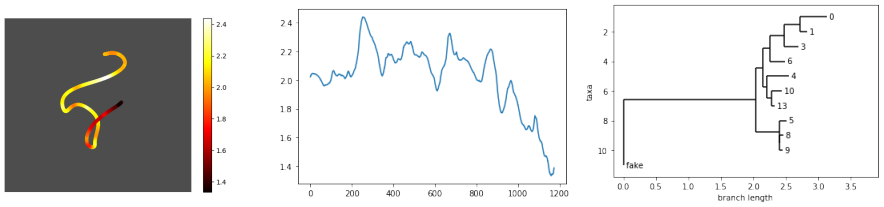
Patients represented in the AneuRisk65 dataset are organized in three groups: the Upper group (U) collects patients with an aneurysm in the Willis circle at or after the terminal bifurcation of the ICA, the Lower group (L) gathers patients with an aneurysm on the ICA before its terminal bifurcation, and finally the patients in the None group (N) do not have a cerebral aneurysm. Our main goal is supervised classification with the aim to develop a classifier able to discriminate membership to the group $L \bigcup N$ against membership to the group $U$ based on the geometric features of the ICA. We complement this supervised analysis with an unsupervised exercise with the aim of clustering patients solely on the basis of the similarity of geometric features of their ICA, recovering a clear structure between the groups listed above and thus providing further support to the




**(a)** *ICA patient 1* **(b)** *Radius along the ICA* **(c)** *Associated tree*




**(d)** *ICA patient 55* **(e)** *Radius along the ICA* **(f)** *Associated tree*




**(g)** *ICA patient 56* **(h)** *Radius along the ICA* **(i)** *Associated tree*

**Figure 3.9:** *Three patients in the AneuRisk65 dataset; on the left column, ICAs are coloured according to the radius value, on the central column the radius functions, on the right column their associated merge trees. Patient 1 belongs to the Lower group, the other two patients to the Upper group.*

discriminating power of the geometric features of the ICA.

**The pipeline for supervised classification**

We develop a classification pipeline in close analogy with the one illustrated in Sangalli et al. (2009b) which, after smoothing, reduces the data dimensionality by means of Functional Principal Components Analysis (FPCA) applied to the curvature functions of the ICA centerlines and to the respective radius functions, and then fits a quadratic discriminant analysis (QDA) based on the first two FPCA scores of the curvature functions and of the radius functions respectively.

We interpolate the data points representing the smoothed curvature functions and the raw radius functions provided in the Aneurisk65 dataset with a piecewise linear spline and, for each patient, we consider the merge tree associated to its curvature and the merge tree associated to its radius. We then prune our tree representations; to use a uniform scale across all patients (but of course different for curvature and radius) we parametrize the pruning threshold as a fraction of the total range covered by the curvature and radius functions, respectively, across patients: $I = [\min_f(\min_x(f(x)), \max_f(\max_x(f(x)))]$. For both sets of trees, we then calculate the pairwise distances with the metric $d_E$ and we organize them in two distance matrices. Blending the discriminatory information provided by curvature and radius, we also produce a new distance matrix collecting the pairwise distances obtained by convex linear combination of the distances for curvature and radius, according to the formula:

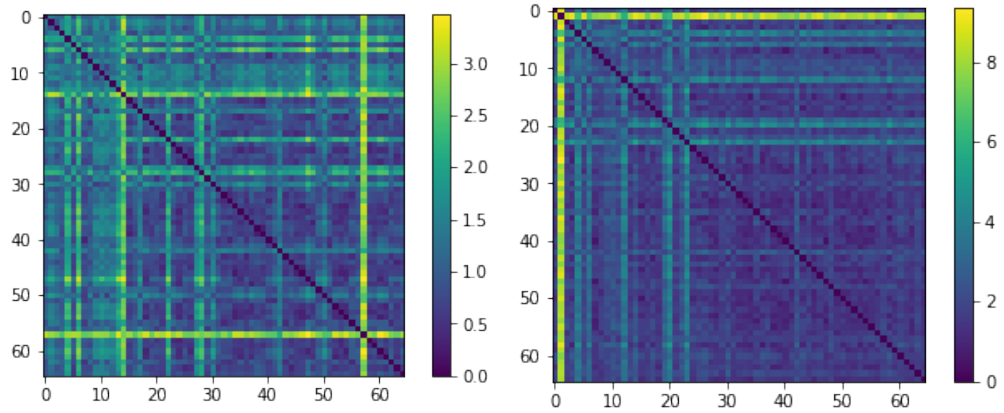
$$d^2_{\text{mixed}} = w \cdot d^2_{\text{curvature}} + (1 - w) \cdot d^2_{\text{radius}}, \tag{3.1}$$

where $0 \leqslant w \leqslant 1$. For lack of references, we prove in Section 3.12.1 that $d_{\text{mixed}}$ is a metric, for all $w \in [0, 1]$. We then apply Multi Dimensional Scaling (MDS) to each of the above distance matrices, to map the results in a finite dimensional Euclidean space of dimension $m$. Lastly, and following Sangalli et al. (2009b), we fit a QDA on such embedded points.

This pipeline requires the setting of three hyperparameters: the pruning threshold, the weight $w$ appearing in Equation (3.1) and, finally, the dimension $m$ of the Euclidean embedding for MDS. While the pruning threshold is chosen with an elbow analysis, see Section 3.9.2, the weight $w$ and the dimension $m$ of the multidimensional scaling are selected by maximising the discriminatory power of QDA estimated by means of leave-one-out (L1out) cross-validation.

**Pruning**

In this section, we take a closer look at the smoothing carried out by pruning the merge trees representations of curvature and radius. From the plots in Figure 3.9 we see that the radius functions appear to be very wiggly and, given the complex data-generating pipeline, we might assume that some portion of that amplitude variability is uninformative and due to different kinds of errors, which is the same conclusion drawn by Sangalli




**(a)** *Pairwise distances between merge trees representing curvatures.*

**(b)** *Pairwise distances between merge trees representing radii.*

**Figure 3.10:** *The distance matrices of merge trees associated to curvature and radius functions.Patients belonging to group $L$ appear in first rows, followed by patients in the $N$ group and patients in the $U$ group.*

et al. (2009a) with respect to the raw curvature data. As detailed in Section 3.6.1, removing those little spikes from functions amounts to removing little branches from trees (up to smaller siblings). Thus, the number of leaves in a pruned tree is a monotone decreasing function of the pruning threshold: as the parameter grows, the number of leaves decreases.

We expect to find some separation in terms of amplitude between the proper features of the analyzed functions and the unwanted, "noisy" ones. Otherwise it would mean that the signal-to-noise ratio is so low that the uninformative errors shadow the informative features of the functions, and thus that the data are hopelessly corrupted. For this reason, we choose the pruning parameter through an elbow analysis of the curve plotting the number of leaves of the pruned trees, averaged over the whole dataset, against the corresponding threshold. Thus, we look for an elbow in the curves depicted in Figure 3.11.

We want here to emphasise the different behaviours of the curvature trees and of the radius trees. There is no clear elbow in Figure 3.11(a), showing that there is no reason to believe that data show a large number of small uninformative noisy oscillations. This is not surprising because the curvature functions of the Aneursik65 dataset are the result of a very careful smoothing process. The curve in Figure 3.11(b), related to radius, has instead a clear elbow structure (between $1\%$ and $2\%$) in accordance with our expectations. Thus, we choose $2\%$ as pruning threshold for the radius curves, whilst we do not prune curvature trees. We later discuss the robustness of our results with respect to these choices.

#### Classification Results

We compare our classification results with those illustrated Sangalli et al. (2009b). The goal is the same: separating the class U from the classes L and N.

Table 3.1 reports the prediction errors obtained after L1out cross-validation. As in Sangalli et al. (2009b), we obtain the best classifier by simultaneously considering the combined information conveyed by the couple of curvature and radius functions; the dissimilarity between different couples is measured by the distance in Equation (3.1), where the parameter $w = 0.46$, being this the value which minimizes prediction error computed by L1out.

The same pipeline is followed when curvature and radius functions are represented by merge trees or by persistence diagrams. In the case of PDs', we first removed the points (that is, the topological features) with persistence lower than a certain threshold (where the persistence is $c_y - c_x$, according to the notation used in Section 3.3). The threshold has been taken equal to the pruning parameter of the merge trees.

The first two rows in Table 3.1 compare prediction errors when merge trees or PD representations are used. From left to right, the table shows the L1out confusion matrices when distances between curvatures, radii or their joint couple are respectively considered. We see that PDs do a better job in extracting useful information from either curvature or radius, when examined separately. This could be due to a situation not dissimilar from that illustrated in the example of Section 3.8.1: the discriminant information contained in the curvature and radius functions lies more in the number and amplitude of oscillations than in their ordering. However, when curvature and radius of the ICA are jointly considered as descriptors and the distance of Equation (3.1) is used, we obtain a better classifier for merge trees while there is no improvement for PDs.

This situation highlights that merge trees and persistence diagrams capture different pieces of information about the represented functions; moreover, PDs suggest that most of the information they capture is due to the radius function, while merge trees clearly show some informative interactions between curvature and radius.

For comparison, the third column of Table 3.1 reports the prediction errors of the best classifiers based on merge trees and on PDs, respectively, while the last row shows the prediction errors of the classifier described in Sangalli et al. (2009b). Although the number of patients misclassified by the best classifier based on merge trees is smaller than that of the best classifier based on PDs, we stress once again that the two methods are capturing different discriminant information; indeed, comparing the two analysis we found that only 6 patients were misclassified by both methods.

#### Robustness with respect to the pruning threshold

To argument in favor of the robustness of our results with respect to the choice of the pruning threshold, or, from another point of view, in favor of the robustness of the information conveyed by our tree representations of functions, we go through the same classification pipeline varying the value of the pruning threshold. In Figure 3.12 we

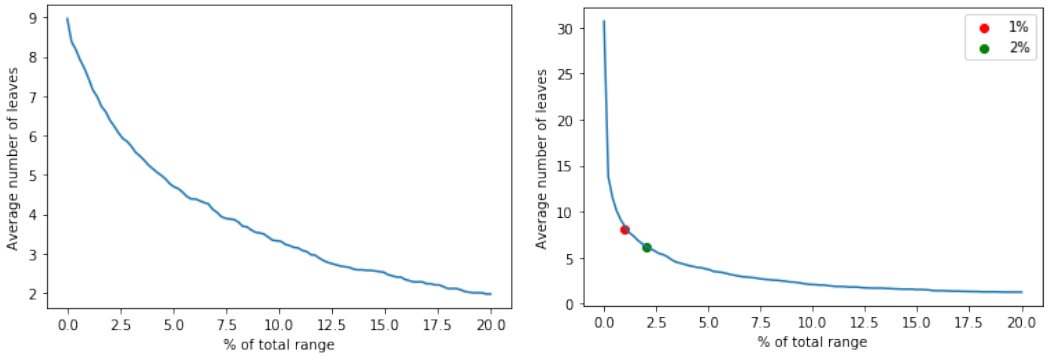
show the prediction accuracy, estimated by L1out cross-validation, as a function of the pruning threshold. We notice that the accuracy is quite stable and, in particular for merge trees, slowly decreases as the threshold increases. This fact, on one hand further supports the elbow analysis approach described in Section 3.9.2, on the other is also showing that the results obtained with the information captured by merge trees and persistence diagrams does not depend on a finely tuned choice of the threshold parameters.

### Clustering

We now explore the Aneurisk65 data clustering structure by endowing the merge tree space with the metric $d_{\text{mixed}}$ figuring in Equation (3.1), with $w = 0.46$. To get multiple perspectives on this issue, we resort to hierarchical clustering dendrograms with different linkages. The visual inspection of Figure 3.10 suggests that, upon blending together the information of radius and curvature, the Upper class should display a low variability while the Lower and None classes should behave more heterogeneously. Thus, a clear clustering structure should not be recognizable: we expect possibly one cluster made by points belonging to the Upper class and then a series of points scattered around this central nucleus with no easily recognizable pattern.

The hierarchical dendrograms obtained with single, average and complete linkages are displayed in Figure 3.13. The first obvious observation is that all three linkages identify the point associated to patient 2 as an outlier. The single linkage dendrogram shows that, as the height on the dendrogram increases, there is only one major cluster which slowly becomes larger and incorporates all points in the data set. No other relevant clusters are found. Average and complete linkages further support this finding: there are no obvious heights where to cut the tree in the average linkage dendrogram; complete linkage instead shows perhaps a two cluster (plus one outlier) structure. The smaller cluster identified by this dendrogram, is also visible with the average linkage and is contained within the group of singletons obtained by cutting the single linkage tree at height $1.3$. The overall picture is thus that of a major cluster, with possibly another group of points clustered together, but with much higher heterogeneity.

These findings can indeed be related with the labels declaring membership of the patients to the $U$, the $N$ and the $L$ group respectively. To grasp if there is an overall pattern in the merging structure of the data point cloud, for each leaf (a patient) of a dendrogram, we collect its merging height defined as the height of its father in the graph, that is the height at which that point is no longer considered as a singleton but instead it is clustered with some other point. In other words, we record the distance between the leaf and the closest cluster in terms of the cophenetic distance induced by the dendrogram. Note that, for the single linkage dendrogram, this is equivalent, for almost all leaves, to the height at which the leaf is merged with the major cluster. Results are shown in Figure 3.14. The interpretation of these plots is consistent across the different linkages and is pretty straightforward: the points corresponding to patients of the Upper group get merged within a small range of heights, and the distribution of their merging height




**(a)** *Average number of leaves for curvature* **(b)** *Average number of leaves for radius*

**Figure 3.11:** *The average numbers of leaves in the merge trees, plotted against the percentage of total range used as pruning threshold.*

is stochastically smaller than the distributions for groups $L$ and $N$, respectively. The merging heights of the leaves corresponding to patients belonging to the Lower group, instead, display a larger variability and their distribution is stochastically larger than those of the of the leaves belonging to the other two groups. Patients of the class None, merge at heights in between the Upper and the Lower groups and their merging height seems to display a low variability. The plot (d) of Figure 3.14 shows the smoothed densities of the distributions of merging height for leaves belonging to the three groups, in the case of average linkage. Analogous representations could be obtained for the other two linkages; they all confirm the stochastic ordering described above.

This cluster analysis is consistent with our expectations, which, in turn, are in accordance to the findings of Sangalli et al. (2009b). On top of that, we also get two further insights: first, data are scattered around the Upper group with a possibly non uniform structure, as shown by the small and sparse cluster of Lower class patients visible with complete linkage clustering and, second, that the None group of patients lies in a sort of in between situation in the space separating the two other groups of aneurysm-affected patients. This could also explain the good performance of QDA: a quadratic boundary is able to isolate the core of the Upper group of patients from the others, which lie mainly on one side of the quadratic discriminant function.

## 3.10 Discussion

We believe that methods from TDA can be fruitfully added to the toolbox of functional data analysis, especially when non trivial smoothing and alignment are required for data representation. In this chapter we focused on two topological representations of functions: persistence diagrams, which, being the most classical tool in TDA, are regarded as

**Merge Trees**

| | | Predicted: Curvature | | Predicted: Radius | | Predicted: Mixed | |
|---|---|---|---|---|---|---|---|
| | | U | L⋃N | U | L⋃N | U | L⋃N |
| True | U | 22 | 10 | 16 | 16 | 25 | 7 |
| | L⋃N | 7 | 26 | 2 | 31 | 3 | 30 |
| | | $w = 1, n = 3$ | | $w = 0, n = 3$ | | $w = 0.46, n = 9$ | |

**Persistence Diagrams**

| | | Predicted: Curvature | | Predicted: Radius | | Predicted: Mixed | |
|---|---|---|---|---|---|---|---|
| | | U | L⋃N | U | L⋃N | U | L⋃N |
| True | U | 21 | 11 | 26 | 6 | 26 | 6 |
| | L⋃N | 3 | 30 | 6 | 27 | L | 6 | 27 |
| | | $w = 1, n = 3$ | | $w = 0, n = 9$ | | $w = 0, n = 9$ | |

**Benchmark**

| | | Predicted | |
|---|---|---|---|
| | | U | L⋃N |
| True | U | 26 | 6 |
| | L⋃N | 6 | 27 |

**Table 3.1:** *Confusion matrices for L1out. Below each confusion matrix, the values of the metric coefficient $w$ and of the dimension $m$ for MDS corresponding to the tested classifier are reported. The first row refers to the classifiers receiving as input merge tree representations, the second row PDs. The last row reports the benchmark L1out confusion matrix for the classifier illustrated in Sangalli et al. (2009b).*

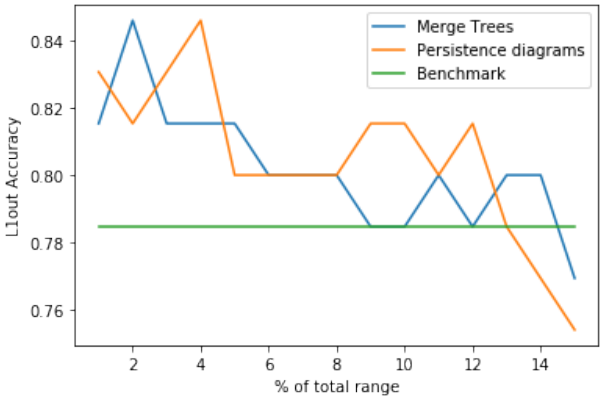
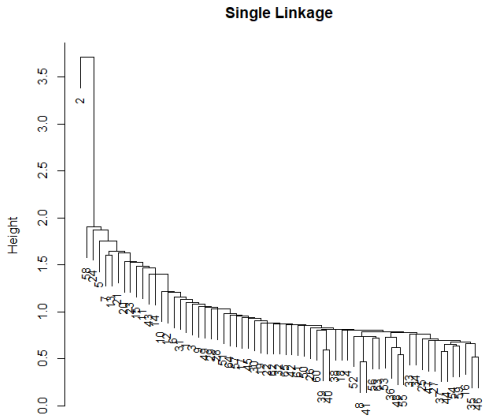



**Figure 3.12:** *We can visually inspect robustness of the L1out accuracy of the classification pipeline - both for merge trees and persistence diagrams - with respect of the pruning threshold. The horizontal green line shows the accuracy obtained by Sangalli et al. (2009b). Note that the accuracy of persistence diagrams and merge trees is above or equal to the green line also for large values of the pruning threshold.*

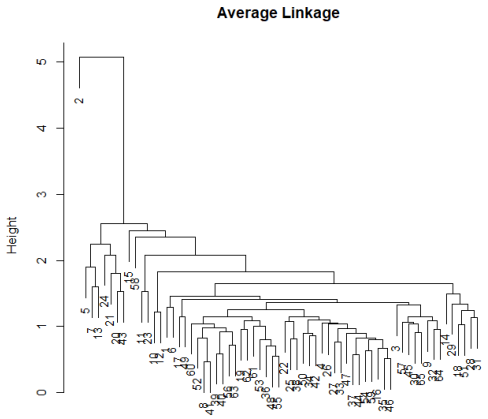
a benchmark, and merge trees, which are rarely used in real data analysis applications. The framework for merge trees is the very recent metric structure defined in Chapter 2, for which we also developed theoretical results specific for the application to functional data.

To support our narrative, we used as paradigmatic real world application the classification analysis of the AneuRisk65 functional data set. This data set poses all the desired challenges: careful smoothing procedures and alignment techniques must be employed to obtain meaningful results. Reanalyzing the seminal case study described in Sangalli et al. (2009b), we show the advantages of having a representation of functional data which is invariant with respect to homeomorphic transformations of the abscissa – thus lightening the burden of careful alignment – and also allows for agile smoothing – possibly causing some overfitting – thanks to the pruning of the trees which takes care of this aspect of FDA which practitioners often find problematic. Following a classification approach based on QDA applied to proper reduced representations of the data, as in Sangalli et al. (2009b), we obtain robust results with comparable, if not better, accuracy in terms of L1out prediction error, and we confirm some facts about the variability of the data in the groups of patients characterized by the different location of the cerebral aneurysm, consistently with the findings of previous works.

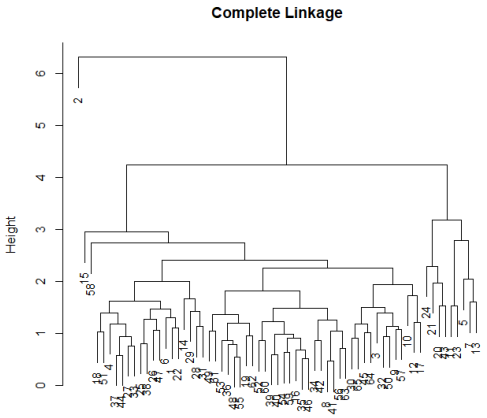
To be sure, we want to stress that careful smoothing is still mandatory when precise differential information about the data is needed, since small oscillations in a function can still cause high amplitude oscillations in the derivatives, which cannot be removed by pruning. Moreover, not all FDA applications are adapted to the representations offered by merge trees or persistent diagrams. Indeed, the information collected by merge trees is contained in the ordering and in the amplitude of the extremal points of a func-




**(a)** *Single linkage clustering dendrogram.*

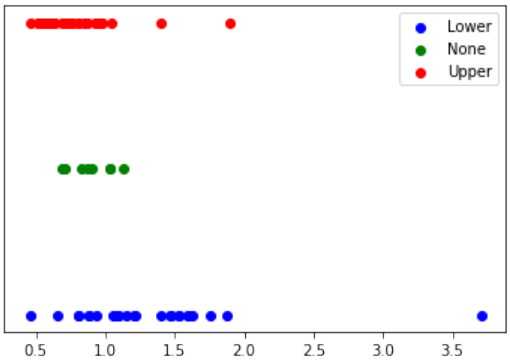



**(b)** *Average linkage clustering dendrogram.*

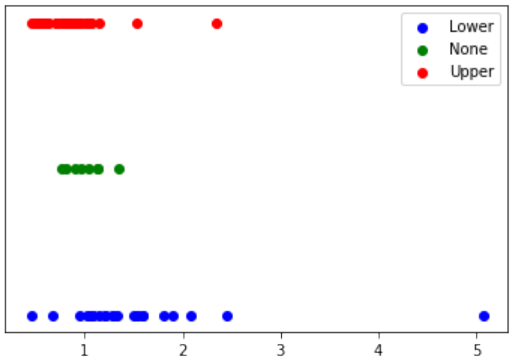



**(c)** *Complete linkage clustering dendrogram.*

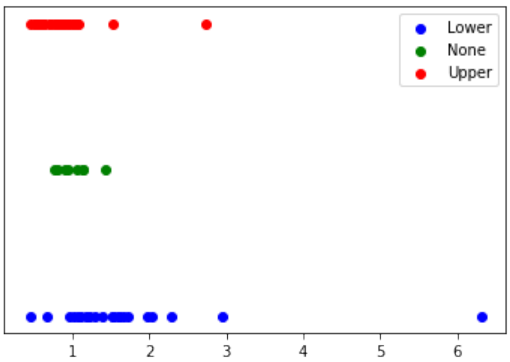
**Figure 3.13:** *Hierarchical clustering dendrograms obtained with single, average and complete linkages.*



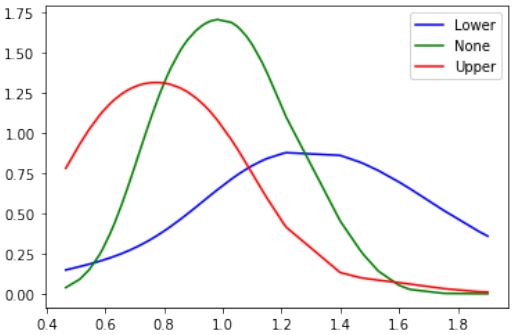

**(a)** *Heights at which leaves get merged in the single linkage clustering dendrogram.*




**(b)** *Heights at which leaves get merged in the average linkage clustering dendrogram.*




**(c)** *Heights at which leaves get merged in the complete linkage clustering dendrogram.*




**(d)** *Density estimate for the distributions in Figure 3.14(b).*

**Figure 3.14:** *For each patient belonging to group $U$, $L$ or $N$, the plots (a), (b) and (c) represent the merging height at which their corresponding leaf gets merged in the clustering dendrograms, according to single linkage, average linkage and complete linkage, respectively. Plot (d) represents the smoothed densities of merging height for the leaves of the three groups, in the case of average linkage.*

tion, and not on their exact abscissa. Hence, if the abscissa carries valuable information for the analysis – for instance, a wavelength, or a precise landmark point in space or time – the TDA approach followed in this work for data representation is not indicated, precisely because of its invariance property with respect to homeomorphic transformations of the abscissa. But this criticism also applies to many alignment procedures proposed in the literature. Similarly, in Section 3.8.2 we point out that there are functions which have equivalent representations in terms of merge trees although the order on the abscissa of their critical points is different, although merge tree are much less sensitive to such issue when compared to persistence diagrams. If the order of critical points of the function is of importance for the analysis, then surely persistence diagrams, but possibly also merge trees, should be avoided.

Going general, we point out that whenever the datum designating a statistical unit is only a representative of an equivalence class, the analyst must be sure that the variability differentiating the members of the same class is ancillary with respect to the statistical analysis performed on the statistical units. This consideration always applies in FDA, whenever data are aligned according to transformations belonging to a group. Merge trees offer a representation of functional data in terms of equivalence classes whose members are invariant with respect to homeomorphic transformations of the abscissa. Persistence diagrams partition the space of functional data in even coarser equivalence classes, although they could be enough for the analysis, as we saw in the case study illustrated in Section 3.9. Occam's razor should guide the analyst's final choice.

## 3.11 Acknoweldgements

We thank Steve Marron who initiated us to algebraic topology for the statistical analysis of functional data during seminal discussions while one of us (PS) was visiting him at UNC.

## 3.12 Proofs

*Proof of* Proposition 5.

Let $f : X \to \mathbb{R}$ be a bounded function defined on a path connected topological space $X$ and let $\varphi : Y \to X$ be an homeomorphism. We need to prove that the merge tree and the persistence diagram associated to the function $f$ and $f' = f \circ \varphi$ are isomorphic.

We know that:

$$Y_t = \{f'^{-1}((-\infty, t])\} = \{y | f'(y) \leqslant t\} = \{x = \varphi(y) | f(x) \leqslant t\}$$

This means that $y \in Y_t$ if and only if $\varphi(y) \in X_t$, and so $Y_t = \varphi^{-1}(X_t)$. In other words, $\varphi$

maps sublevel sets into sublevel sets. Since the restriction of an homeomorphism is still an homeomorphism, it also sends connected components into connected components.

As a consequence $H_0(X_t) \simeq H_0(Y_t)$ and if $\{x_0, ..., x_n\}$ is a basis made by connected components for $H_0(X_t)$ and $\{y_0, ..., y_n\}$ for $H_0(Y_t)$, $\varphi$ induces the following isomorphism of groups on the homologies: $\widetilde{\varphi} : H_0(X_t) \to H_0(Y_t)$, such that $\widetilde{\varphi}(x_i) = y_j$ for some $j$.

Given $t' < t$, we have the following commutative diagram:

$$\begin{array}{ccc} X_{t'} & \longrightarrow & X_t \\ \downarrow{\scriptstyle \varphi} & & \downarrow{\scriptstyle \varphi} \\ Y_{t'} & \longrightarrow & Y_t \end{array}$$

and passing to homologies:

$$\begin{array}{ccc} H_0(X_{t'}) & \xrightarrow{\alpha_X} & H_0(X_t) \\ \downarrow{\scriptstyle \widetilde{\varphi}} & & \downarrow{\scriptstyle \widetilde{\varphi}} \\ H_0(Y_{t'}) & \xrightarrow{\alpha_Y} & H_0(Y_t) \end{array}$$

where we remark that the $\varphi$'s are homeomorphisms and the $\widetilde{\varphi}$'s are isomorphisms of groups.

The last diagram gives the isomorphism of $PD_0(f)$ and $PD_0(f')$.

Since connected components are sent to connected components (with both maps), this also becomes an isomorphism of merge trees: building the merge tree for $Y_t$ and $X_t$ would give isomorphic results. ■

*Proof of* Proposition 6.

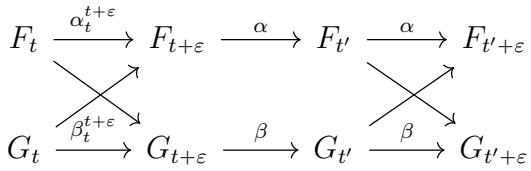
Each leaf in $T$ corresponds to a point in $PD(f)$. The $x$ coordinate of each point is given by its height, which can be retrieved through $h_f$. Consider $v \in L_T$ and $\gamma_v$, the path from $v$ to $r_T$ which corresponds to the ordered set $\{v' \in V_T | v' \geqslant v\}$. The $y$ coordinate of the points associated to $v$ is the minimal height at which $\gamma_v$ intersects $\gamma_l$, with $l$ being a leaf with height less than $v$. ■

*Proof of* Theorem 3.

To prove the theorem, we need some notation and a couple of Lemmas.

Let $f, g$ be tame functions on the path connected topological space $X$ and such that $\sup_{x \in X} |f(x) - g(x)| \leqslant \varepsilon$. For $t \in \mathbb{R}$, we set $X_t^f = f^{-1}((-\infty, t])$. Since $|f(x) - g(x)| \leqslant \varepsilon$ we have $X_t^f \subset X_{t+\varepsilon}^g$ and of course $X_t^g \subset X_{t+\varepsilon}^f$.

If $F_t := \{\text{path connected components of } X_t^f\}$ and, analogously, $G_t := \{\text{path connected components of } X_t^g\}$, we induce the following commutative diagram:




We call $\widehat{\varphi} : F_t \to G_{t+\varepsilon}$ and $\widehat{\gamma} : G_t \to F_{t+\varepsilon}$. Note that the vertices of the merge trees associated to $f$ and $g$ are contained in some $F_t$ or $G_t$ respectively. The maps are all induced by inclusion and are defined in analogous ways on the basis of path connected components (p.c.c.). So we specify only $\widehat{\varphi}$:

$$(U \subset X_t^f \text{ p.c.c. }) \mapsto (V \subset X_{t+\varepsilon}^g \text{ p.c.c. containing } U)$$

To define $\widehat{\gamma}$ we simply exchange the role of $f$ and $g$.

We indicate the path connected components of $F_t$ with $F_0^t, ..., F_k^t$ and analogously for $G_t$. Since $f$ and $g$ are tame, we know that these are always finite. Clearly, we have that $\alpha_t^{t'}(F_i^t) = F_j^{t'}$ for some $j$. Usually we avoid the subscript and the superscript on $\alpha$ referring to its domain and codomain; if needed, we specify them.

If $\alpha_t^{t'}(F_i^t) = \alpha_t^{t'}(F_h^t) = F_j^{t'}$ then $F_i^t$ and $F_h^t$ have merged between $t$ and $t'$. Similarly if $F_j^t \notin im(\alpha_{t'}^t)$ then $F_j^t$ is born between $t'$ and $t$. Since $f$ and $g$ are tame, we know that merging and the birth of new components happens only in a finite set of critical points.

We recall that the leaves of the trees are associated to the birth of path-connected components, and the internal vertices of the trees to the points where components merge.

Given a path connected component $x \in F_t$, we can always find $t' = min\{s \leqslant t | \#\alpha^{-1}(x) = 1\}$; we call $\Gamma_f(x)$ the preimage of $x$ with $\alpha_{t'}^t$. It is the closest point on the tree, going towards the leaves, in which $x$ is involved in some merging. It is a way to associate to any component alive at time $t$, a vertex on the tree. In other words the function $\Gamma_f$ maps any connected component of $F_t$ (for any $t$), to a vertex of the tree $T_f$. An analogous map can be defined also for $G_t$ and $T_g$; call this functions $\Gamma_g$.

Having set notation, we now use it to establish some connections with merge trees as defined in Section 3.2.5.

Let $T_f$ be the merge tree associated to $f$, with tree structure $T$ and height function $h_f$; similarly $T_g$ is associated to $g$, with tree structure $T'$ and height function $h_g$. Define the functions $\varphi$ and $\gamma$, respectively by considering: $F_i^t \mapsto \Gamma_g(\widehat{\varphi}(F_i^t))$ or $G_i^t \mapsto \Gamma_f(\widehat{\gamma}(G_i^t))$. We will mainly use these functions restricted to the sets of vertices $V_T$ and $V_{T'}$, to obtain

$$\varphi : V_T \to V_{T'}$$

$$v \mapsto \Gamma_g(\widehat{\varphi}(v))$$

$$\gamma : V_{T'} \to V_T$$

$$w \mapsto \Gamma_f(\widehat{\gamma}(w))$$

We now prove two lemmas which help in the proof of the theorem. We also introduce the following notation: given a vertex $v \in V_T$, then, the set $\zeta_v$ is the set $\zeta_v := \{v' \in v_T \big| v' \geqslant v\}$. In other words $\zeta_v$ collects all the points between $v$ and $r_T$.

**Lemma 2.** *Consider $v \in V_T$. If $|h_f(v) - h_g(\varphi(v))| > \varepsilon$, then there cannot be two or more distinct vertices $v'$ and $v''$ such that $v = min\ \zeta_{v'} \cap \zeta_{v''}$ with $h_f(v) - h_f(v') > \varepsilon$ and $h_f(v) - h_f(v'') > \varepsilon$. Moreover, for all vertices $v'''$ in $sub_T(v)$ with $h_T(v) - h_T(v''') < \varepsilon$ we have $\varphi(v''') = \varphi(v)$.*

*Proof.* We prove this lemma by contradiction. Assume there are $v'$ and $v''$ vertices which contradict the thesis.

Note that $\widehat{\varphi}(v')$ and $\widehat{\varphi}(v'')$ are less than or equal to $\varphi(v)$ in $V_{T'}$. But by hypothesis $h_g(\varphi(v)) < h_f(v) - \varepsilon$, which means that $h_f(\widehat{\gamma}(\varphi(v))) < h_f(v)$. Which is a contradiction because the components associated to $v'$ and $v''$ cannot merge before $h_f(v)$.

The last part of the lemma follows because $h_g(\widehat{\gamma}(v''')) > h_g(\varphi(v))$.

□

**Lemma 3.** *Consider $F_i^t$ and $F_j^{t'}$ with $t < t'$ and $\varphi(F_i^t) = \varphi(F_j^{t'})$. Then $F_j^{t'}$ and $F_j^{t'}$ get merged before height $t' + 2\varepsilon$.*

*Proof.* If $\varphi(F_i^t) = \varphi(F_j^{t'})$ then $\widehat{\varphi}(F_i^t) \subset \widehat{\varphi}(F_j^{t'})$. This, in turn, implies that $\widehat{\gamma}(\widehat{\varphi}(F_i^t)) \subset \widehat{\gamma}(\widehat{\varphi}(F_j^{t'}))$ and so $F_i^t$ and $F_j^{t'}$ get merged before height $t' + 2\varepsilon$. □

Clearly the role of $T$ and $T'$ can be exchanged in the formulation of the lemmas.

Using these two lemmas we build a bottom-up procedure to turn $T$ into $T'$ via an edit path with at most one edit per vertex, each with cost less then $2\varepsilon$.

Start from the leaves of $T$ and order them according to increasing heights. If there are more leaves with the same height, order them at random. For each leaf $v$ either (a) $|h_f(v) - h_g(\varphi(v))| > \varepsilon$ or (b) $|h_f(v) - h_g(\varphi(v))| \leqslant \varepsilon$ holds.

Consider the first leaf $v$. Since $|\min(f) - \min(g)| < \varepsilon$ then (b) must hold. Thus we take the couple $(v, \varphi(v))$. Consider now the second leaf $v'$. If (a) holds, then $\varphi(v') = \varphi(\widehat{\gamma}(\varphi(v')))$ and $h_f(v') > h_f(\widehat{\gamma}(\varphi(v')))$. Since $v'$ is a leaf, then $\Gamma_f(\widehat{\gamma}(\varphi(v')))$ belongs to another branch, with respect to $v'$, since $h_f(v') > h_f(\widehat{\gamma}(\varphi(v')))$. Thus, by Lemma 3, $v'$ can be deleted with cost at most $2\varepsilon$. The same happens if $\varphi(v') = \varphi(v)$. Therefore, for each leaf $v$ of $T$ either we take the couple $(v, \varphi(v))$ - if (b) holds, and with vertices appearing in couples at most once - or we delete $v$ with cost less than $2\varepsilon$. Consider now the leaves of $T'$. If $w$ is left uncoupled and (a) holds, then $\gamma(w) = \gamma(\widehat{\varphi}(\gamma(w)))$ with $w > \widehat{\varphi}(\gamma(w))$; reasoning as above, we deduce that $w$ can be deleted with cost at most $2\varepsilon$. If (b) holds, then $|h_g(w) - h_g(\varphi(\gamma(w)))| < 2\varepsilon$, since $|h_f(\gamma(w)) - h_g(\varphi(\gamma(w)))| < \varepsilon$. Thus we can delete $w$ with cost less than $2\varepsilon$ in any case.

Therefore, we either couple or delete each leaf of $T$ and $T'$.

These deletions may force some vertices to become leaves. Thus we can repeat recursively the same procedure until we obtain two merge trees whose leaves are all

coupled. From such trees we remove all order two vertices. Since these trees are obtained from $T$ and $T'$ with deletions of cost less than $2\varepsilon$ and ghostings, if we prove the result for such trees the theorem is proven. So with an abuse of notation we call $T$ and $T'$ these new merge trees.

To conclude the proof we must first prove that the internal vertices of $T$ and $T'$ satistfy (b) and can be coupled respecting the tree structures.

Consider $v$ an internal vertex of $T$ and suppose $|h_f(v) - h_g(\varphi(v))| > \varepsilon$. Let $v_1, \dots, v_n$ be its children. By hypotesis $n > 1$. We know that all the leaves in $sub_T(v_i)$ are coupled. In particular consider two leaves $v_a \in V_{sub_T(v_1)}$ and $v_b \in V_{sub_T(v_2)}$. We know that $\varphi(v_a) \neq \varphi(v_b)$, and that those two components are merged in $\varphi(v)$. But then they are merged in $\widehat{\gamma}(\varphi(v))$ which has height less then $h_f(v)$; a contradiction. This of course holds also for $T'$.

Consider now the set $\varphi^{-1}(\varphi(v)) = \{v_1, \dots, v_n\}$. We know that for all $i$, $|h_f(v_i) - h_g(\varphi(v))| < \varepsilon$ and all the vertices get merged before $\max_i\{h_f(v_i)\} + \varepsilon$. Let $k = \text{argmax}_i\{h_f(v_i)\}$. We can pair $(v_k, \varphi(v))$ and delete all other $v_i$, with cost less than $2\varepsilon$. In this way we either couple or delete all vertices of $T$. Consider now $w \in V_{T'}$ which is left uncoupled. Since for $w$ (b) holds, then $|h_g(w) - h_g(\varphi(\gamma(w)))| < 2\varepsilon$, because $|h_f(\gamma(w)) - h_g(\varphi(\gamma(w)))| < \varepsilon$. Thus we can delete all uncoupled vertices with deletions whose cost is less than $2\varepsilon$.

In this way all vertices of $T$ are either coupled, ghosted or deleted. Lastly, since $\widehat{\varphi}(v) \geqslant \widehat{\varphi}(v')$ then the coupling respects the tree structures of $T$ and $T'$. Therefore, once we delete all vertices which are not coupled, and remove all order 2 vertices, we obtain from $T$ and $T'$ two trees with isomorphic tree structures - the isomorphism being $\varphi$. Again to avoid the introduction of new notation, we call these trees $T$ and $T'$. At this point we can intepret the couples $(v, \varphi(v))$ as defining shrinkings. Let $e = (v, v')$, $w_T(e) = h_f(v') - h_f(v)$, $e' = (\varphi(v), \varphi(v'))$ and $w_{T'}(e') = h_g(\varphi(v), \varphi(v'))$. Since $|h_f(v) - h_g(\varphi(v))| < \varepsilon$ and $|h_f(v') - h_g(\varphi(v'))| < \varepsilon$, then $cost((v, \varphi(v))) = |w_T(e) - w_{T'}(e')| < 2\varepsilon$.

■

### 3.12.1 Combining Metrics

To aggregate curvature and radius, we make use of the following Proposition.

**Proposition 7.** *Given $(X, d_0)$ and $(X, d_1)$ metric spaces, then $d_{a,b,p} := (a \cdot d_0^p + b \cdot d_1^p)^{1/p}$, with $a, b \in \mathbb{R}_{>0}$ and $p \geqslant 1$, is a metric on $X$.*

*Proof.* $d_{a,b,p}(x, y) = ||(a^{1/p} \cdot d_0(x, y), b^{1/p} \cdot d_1(x, y))||_p$.

Since, given $k > 0$, $k \cdot d_i$ is a metric if and only if $d_i$ is a metric, we can rescale $d_0$ and $d_1$ and take $a = b = 1$. We refer to $d_{1,1,p}$ as $d_p$.

So:

- $d_p(x, y) = 0$ iff $d_0(x, y) = 0 = d_1(x, y)$ and this happens if and only if $x = y$.

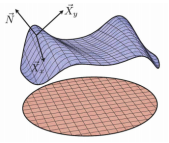
- symmetry is obvious
- we use $||h+q||_p \leqslant ||h||_p+||q||_p$ with $h = (d_0(x,z), d_1(x,z))$ and $q = (d_0(z,y), d_1(z,y))$.
  Since $d_i(x,y) \leqslant d_i(x,z) + d_i(z,y)$ we get:
  $||(d_0(x,y), d_1(x,y))||_p \leqslant ||(d_0(x,z)+d_0(z,y), d_1(x,z)+d_1(z,y))||_p = ||(d_0(x,z), d_1(x,z))+(d_0(z,y), d_1(z,y))||_p \leqslant ||(d_0(x,z), d_1(x,z))||_p + ||(d_0(z,y), d_1(z,y))||_p$.

  Therefore:
  $d_p(x,y) \leqslant d_p(x,z) + d_p(z,y)$.

□

CHAPTER 4

# The Space of Merge Trees



The content of this chapter is also part of the paper: "Frechét Means of Finite Sets of Merge Trees" which is at a preliminary stage.

In this chapter we consider possibly the simplest interesting case of generalized dendrograms, which is the one of merge trees, and we start an investigation of the properties of such space. This investigation is intended as a first step into a geometric understanding of the spaces of generalized dendrograms, which is fundamental for developing exploratory tools and other differential structures. In particular, we call $(\mathcal{T}, d_E)$ the metric space of merge trees identified up to order 2 vertices and present some results about its topological properties and its metric structure, with particular attention to objects called Frechét Means.

## 4.1 Preliminaries

In this chapter we make use of the following mathematical structures: categories and multivalued functions.

Category theory is a very big field in mathematics and there is plenty of introductory textbooks which can be used to acquire the basic notions we employ in this part of the dissertation. Namely these notions are the definition of a category, of functors and natural transformations, of partially ordered sets (posets) seen as categories and lastly of coproducts, which can be found for instance in Borceux (1994).

Multivalued functions instead are much more basic objects. Given two sets $A$ and $B$, a multivalued function is an association rule $f : A \to B$ such that $f(a) \subset B$.

## 4.2 Subspaces

The first structure we notice in the space $(\mathcal{T}, d_E)$ is that it can be stratified, covered with a nested family of subsets of merge trees grouped according to the dimension of their representative without order 2 vertices. If we denote with $dim(T)$ the number of edges in the tree structure $T$, we can give the following definition.

**Definition 20.** $\mathcal{T}^N = \{T | dim(T_2) \leqslant N\}$ *for any* $N \in \mathbb{N}$.

**Remark 16.** *Throughout Section 4.2, Section 4.3 and Section 4.4, unless specified otherwise, we always assume that* $T = T_2$ *for any merge tree we consider.*

The results proved in Chapter 2, tells us that for any pair of trees the distance between them is given by the length of a path connecting them. Such path is a geodesic. Moreover, looking at how mappings parametrize finite edit paths, we see that if $T, T' \in \mathcal{T}^N$, then there is at least a geodesic between them which does not exit $\mathcal{T}^N$. We sum up these things with the following proposition.

**Proposition 8.** $(\mathcal{T}, d_E)$ *and* $(\mathcal{T}^N, d_E)$ *are geodesic spaces.*

Understanding how these strata interact with each other and how we can navigate between them can shed some light on the structure of the space $(\mathcal{T}, d_E)$.

## 4.3 Topology

Topology plays a central role when investigating the properties of a space. For instance, being able to characterize or identify open, closed and in particular compact sets is fundamental to work with real valued operators defined on such space.

Firstly we observe that the reversed triangle inequality in the case of generalized dendrogram spaces has the following form.

**Proposition 9.** *Given $T$, $T'$ generalized dendrograms, then:*

$$\big|||T|| - ||T'|||\big| \leqslant d_E(T, T')$$

The following result presents some topological properties of the space $\mathcal{T}$ and its subspaces $\mathcal{T}^N$.

**Theorem 4.** *For any $N \in \mathbb{N}$:*

1. *$(\mathcal{T}, d_E)$ is a contractible geodesic space.*
2. *$(\mathcal{T}, d_E)$ is not locally compact.*
3. *$(\mathcal{T}^N, d_E)$ is a locally compact, contractible geodesic space.*

Theorem 4 states that our spaces are "without holes", that is we can continuously shrink the whole space onto the tree with one vertex and no edges and so $\mathcal{T}$ and $\mathcal{T}^N$ are contractible. As predictable $\mathcal{T}$ has at every point issues with losing compactness because of the growing dimension of the trees, issues which can be solved by setting an upper bound on the dimension, which means working in $\mathcal{T}^N$ for some $N$. Thanks to these results we can further characterize the subspaces $\mathcal{T}^N$, with the following results.

**Theorem 5.** *$(\mathcal{T}^N, d_E)$ is a complete metric space.*

Completeness is a fundamental property, which if very important if one wants to achieve some kind of compactness inside the space of interest. In fact there are many sufficient conditions for compactness, when the completeness of a space is proven.

**Proposition 10.** *The set $||T|| < C$ in $(\mathcal{T}^N, d_E)$ is complete and totally bounded.*

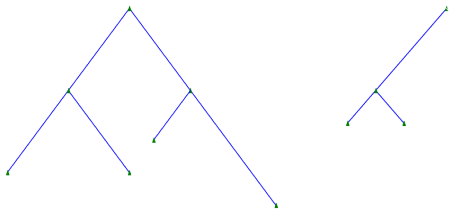
Since a metric space which is complete and totally bounded is compact, we have the following Corollary which tells us that, from the compactness point of view, when we bound the dimension of the trees we are not far away from the behaviour of $\mathbb{R}^N$.

**Corollary 4.** *The set $||T|| < C$ in $(\mathcal{T}^N, d_E)$ is compact.*

As a consequence, whenever we can bound the norm of some merge trees and their dimension, we know that we are moving inside a compact set.

## 4.4 Metric structure

When working outside linear spaces there are many definitions that must be reinterpreted and generalized to work where no linear structure is available. In the case of manifold, the most common way to do so is exploiting locally the linear structure of the tangent space and to focus on the geodesic nature of straight lines in linear spaces. For



**(a)** $T$ **(b)** $T'$

**Figure 4.1:** *With the weights properly defined, it is clear that the tree $T'$ can be mapped both in the left and in the right subtrees of the root $r_T$ with equal cost.*

instance in Geodesic Principal Component analysis (Huckemann et al., 2010a), principal components are replaced by geodesic minimizing the average distance from data points and orthogonality if verified in the tangent space at the barycenter.

For this reasons we want to get a better understanding of the metric structure of the tree space, with particular attention to its geodesic paths.

**Proposition 11.** *For every $T \in \mathcal{T}$, for every $\varepsilon$, exists $T' \in \mathcal{T}$ such that $T$ and $T'$ are connected by multiple minimizing mappings and $d_E(T, T') < \varepsilon$.*

For example consider the trees in Figure 2.5. Suppose $w(A) = w(B) < \varepsilon/2$ and $w(A') = w(C)$; then both $M = \{(B, D), (C, G), (A, A')\}$ and $M' = \{(A, D), (B, D), (C, A')\}$ are minimizing mappings with costs less than $\varepsilon$. We can replicate this situation for any leaf of any trees. The set of points with non-unique minimizing mappings (and so geodesics) is therefore the whole space.

There are two reasons for this non uniqueness to arise:

- similarity between subtrees of the same tree;
- exchange of father-son relationships through the deletion of internal edges.

We can see in Figure 4.1 a more general example of non uniqueness because of similar subtrees, and also in the proof of Proposition 11 we see this problem in action between subtrees made by a branch each.

In Figure 4.2, on the other hand, we can see uniqueness being broken by topological changes made with internal edges: if we need to change lengths of branches sometimes it can be less expensive to make topological changes like deleting internal edges, and regrowing them to swap children. When this kind of mapping is as expensive as adjusting the children we have of course multiple mappings.

To hope to achieve some kind of general uniqueness for mappings we must therefore prevent these things to happen.

Given a merge tree $T$, we use/recall the following pieces of notation:

- the vertices with no children are called leaves ($L_T$);

- vertices which are not leaves are called internal ($I_T$);
- for any vertex $v \in V_T$, $sub(v)$ is the subtree of $T$ rooted in $v$;
- $sub(T) = \{sub(v) | v \in V_T\}$;
- let $child(v) = \{v_1, ..., v_n\}$ be the children of $v$, then $sub_i(v) = sub(v_i) \cup (v_i, v)$, rooted in $v$;
- $Sub(v) = \{sub_i(v) | v_i \in child(v)\}$;
- $dim(T)$ is the number of edges in the tree $T$.

Using this notation, define:

$$k_v = min_{T',T'' \in Sub(v), T' \neq T''} d_E(T', T'')$$

and similarly:

$$k_T = min_{v \in V_T - L_T} k_v$$

.

Lastly let $m_T = min_{v \in V_T - L_T} w(v)$ and $K_T = min\{m_T, k_T\}$.

We want to prove that for trees with $K_T > 0$, if we don't go too far, at least on internal vertices the minimizing mappings are uniquely determined. But we need some preliminary results and tools.

Using the partial ordering induced on the coupled points (seen in Section 2.7), any mapping can be restricted to a subtree rooted in a point $v$ which is neither deleted nor ghosted: in fact if $(v, v') \in M$, $M \in Mapp(T, T')$, then restricts to $M_{|sub(v)} \in Mapp(sub(v), sub(v'))$. If $v$ has some children $v_i$, consider $sub_i(v)$. Then $M$ sends any non deleted $sub_i(v)$ into one or more of $sub_j(v')$; thus, upon adding the trivial subtree (one vertex, no edges) to $Sub(v)$ and $Sub(v')$, we can induce a multivalued function between $Sub(v)$ and $Sub(w)$, which associates subtrees according to $M$ and sends the deleted subtrees into the trivial tree. Note that $sub_i(v) \mapsto \{sub_j(w), sub_k(w)\}$ means that there are deletions of internal edges in $sub_i(v)$ which allows two of its subtrees to be matched one with $sub_j(w)$ and one $sub_k(w)$. The others are deleted. So in the previously defined multivalued function the associations we have are either one-to-one, one-to-many and many-to-one. Thus we can make the multivalued function a bijective function by pinching together set of subtrees at their roots.

For example, suppose $\{V, V'\} \mapsto \{W\}$, then pinch together $V, V'$ and form the tree $\overbrace{VV'}$ which is sent in $W$; similarly if $V, V', V''$ are deleted we map: $\{\overbrace{VV'V''}\} \mapsto \{\varnothing\}$. We want to pinch as few merge trees as possible to make all the associations one-to-one, so subtrees which are already mapped one to one are not pinched. For example if $\{V\} \mapsto \{W\}$ and $\{V'\} \mapsto \{W'\}$ we do not take $\{\overbrace{VV'}\} \mapsto \{\overbrace{WW'}\}$, but keep the

two single associations. This function is called: $\Gamma_M^{(v,w)} : Sub(v) \to Sub(w)$ where, with an abuse of notation we identify $Sub(v)$ and $Sub(v')$ with the sets of pinched subtrees which turn the multivalued function into a function.

One obvious consequence is that the cost of a mapping $M$ can be calculated subtree by subtree using $\Gamma_M^{(r_T,r_{T'})}$; so for instance if $M$ is a geodesic mapping between $T$ and $T'$, $cost(M) = \sum_{V \in Sub(r_T)} d_E(V, \Gamma_M^{(r_T,r_{T'})}(V))$.

**Lemma 4.** *Given a tree $T$, let $Sub(r_T) = \{V_1, ..., V_n\}$. Suppose it exists $\varepsilon > 0$ such that for every $i \neq j$, $d_E(V_i, V_j) > \varepsilon$ and $min_{v \in V_T} cost(v_d) > \varepsilon$. Then, for $\{i_1, \ldots, i_k\}$ and $\{j_1, \ldots, j_h\} \subset \{1, 2, ..., n\}$, if $\overbrace{V_{i_1}, ..., V_{i_k}} \neq \overbrace{V_{j_1}, ..., V_{j_h}}$ then:*

$$d_E(\overbrace{V_{i_1}, ..., V_{i_k}}, \overbrace{V_{j_1}, ..., V_{j_h}}) > \varepsilon$$

Using this Lemma we can start to build up some characterizations of the mappings which parametrize geodesics for $d_E(T, T') < K_T$.

**Remark 17.** *In what follows sometimes we need to consider "sequences" of edges. First notice that, given a tree structure $T$, being $V_T$ partially ordered, then by $E_T \simeq V_T - \{r_T\}$, also $E_T$ inherits a partial order structure. The sequences of edges we consider are always sequences of adjacent ordered edges $\{e_1, \ldots, e_n\}$. It means that we have $e_1 < \ldots < e_n$ and that $e_i$ and $e_{i+1}$ share a vertex. Sometimes, for the sake of simplicity, we just refer to such sequences as sequences of adjacent edges, omitting the ordered property. We may refer to one such sequence of edges as $[v, v']$, meaning a sequence which starts in the vertex $v$ and ends with the vertex $v'$, with $v < v'$.*

**Corollary 5.** *Let $d_E(T, T') < K_T$ and $M$ and $M'$ minimizing mappings. We have $\Gamma_M^{(r_T,r_{T'})} = \Gamma_{M'}^{(r_T,r_{T'})}$*

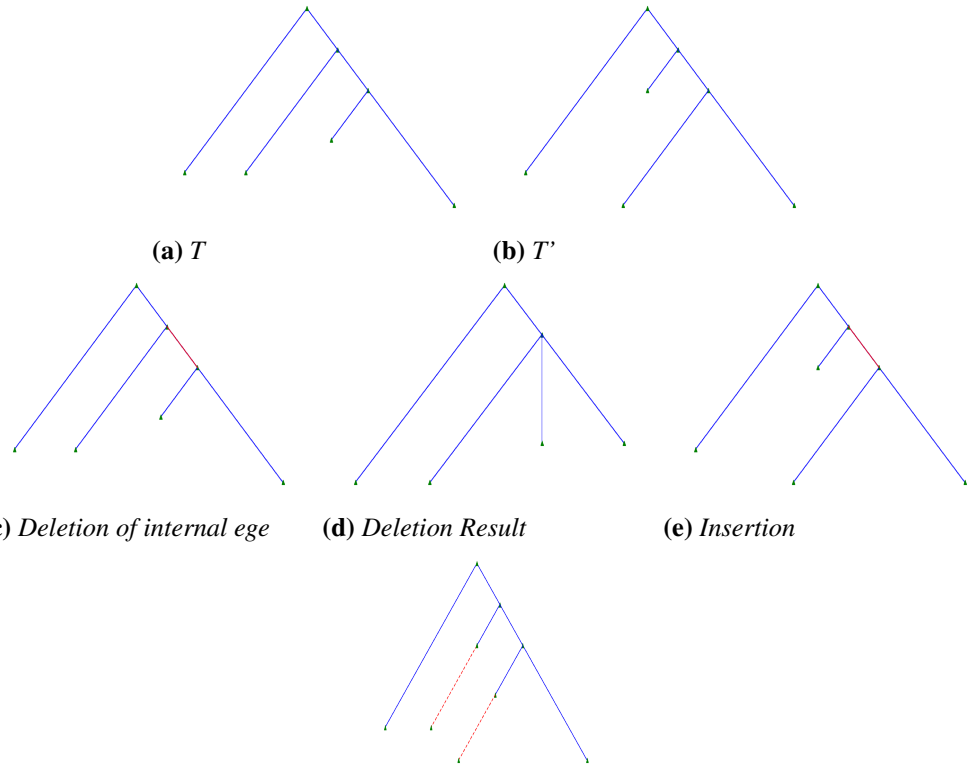
**Lemma 5.** *Let $d_E(T, T') < K_T$ and $M$ and $M'$ minimizing mappings. There cannot be any edge (or sequence of edges) of $T'$ that goes from coupled to a single edge of $T$ in $M$, to deleted in $M'$ (or viceversa).*

*Moreover the deletions on $T'$ shared between $M$ and $M'$, turn $T'$ to a tree, whose representative without order two vertices has the same tree structure of $T$ (up to isomorphisms).*

Using Lemma 5 we can prove the following Corollary, which must old otherwise Lemma 5 is contradicted.

**Corollary 6.** *Let $d_E(T, T') < K_T$ and $M$ and $M'$ minimizing mappings and let $w$ be an internal vertex of $T'$.*

1. *If $sub(w)$ or $sub_j(w)$ is deleted by $M$, then is deleted also by $M'$.*
2. *If after the deletions in $M$, $w$ is of order 2, the same holds after the deletions in $M'$.*



**(a)** *T* **(b)** *T'*

**(c)** *Deletion of internal ege* **(d)** *Deletion Result* **(e)** *Insertion*

**(f)** *Shrinking path*

**Figure 4.2:** *We see in the first line the two merge trees $T$ and $T'$, which admit a minimizing path through deletion and insertion of an internal edge, and a minimizing path which just shrinks the edges. The first path from $T$ to $T'$ is represented, from left to right, in the second line. The third line represents the path made only by shrinkings.*

Finally we can prove the following Theorem.

**Theorem 6.** *If $d_E(T, T') < K_T$, then there is a unique way to define a minimizing mapping on internal vertices.*

## 4.5 Frechét Means

In this section we take the next step in the study of the space $(\mathcal{T}, d_E)$, focusing on the Frechét means of a set of trees.

Frechét means are objects of particular interest in data analysis. They are defined as the minimizers of operators which look for central points in the distribution of a random variable. More formally, given $X$ random variable with values in $(M, d_M)$

metric space, a $p-$Frechét mean is defined as $\text{argmin}_{q\in M}\mathbb{E}_X(d_M(q,x)^p)$ - if it exists. Often this definition is given with $p = 2$ but, at this point we have no reasons to make this choice. As generalization of the idea of "average", or 0-dimensional summary of a random variable, Frechét means are among the most used statistics and data analysis for manifold valued data (Davis, 2008; Pennec, 2006) but not only (Calissano et al., 2020; Turner et al., 2014), and are used as starting points to build more refined tools.

**Proposition 12.** *Given $T_1,\ldots, T_n$ merge trees and $p > 0$; then exist at least one $\overline{T}$ such that:*

$$\overline{T} = \textit{argmin}_T \sum_i d_E^p(T, T_i)$$

Thus, for any finite set of trees, we can minimize the function $T \mapsto \sum_i d_E^p(T, T_i)$, obtaining a $p$-Frechét mean of the subset. We also make the following claim which is still to be investigated.

**Claim 1.** *Given $T_1,\ldots, T_n$ merge trees, if $T_i \in \mathcal{T}^N$, then $\overline{T} \in \mathcal{T}^N$.*

This claim is supported by the fact that $\mathcal{T}^N$ are geodesic spaces, and thus is reasonable that we do not need to increase the dimension to find a Frechét mean.

## 4.6 Tangent spaces and geodesics decomposition

To try to get further insights into the metric structure of the space of merge trees we want to setup some sort of vector calculus for this space. For differential manifolds vector calculus is defined by attaching to a point a linear space parametrizing the possible velocities of curves going through that point. The linear structure of such space can be exploited as a local approximation of the manifold to carry out some linear operations and then map the result back to the manifold. If we look at things the other way round, tangent velocity vectors help understanding how the manifold behaves close to the tangent point. And this is the perspective with which we develop a notion of tangent space in the space of merge trees $(\mathcal{T}, d_E)$: we try to parametrize the possible directions which can be taken starting from a merge tree $T$. In this way we hope to get a better understanding of the local behaviour of the space $(\mathcal{T}, d_E)$.

Throughout the remaining sections of the chapter we need to jump back and forth between merge trees and merge trees identified up to order 2 vertices. To be more careful with the notation, in this section we use $T$ to indicate a merge tree and $[T]$ to indicate its equivalence class up to order 2 vertices. Differently from the previous sections, we are no more assuming $T = T_2$.

We also make use of the following construction. Given a finite set $A = \{a_1, \ldots, a_n\}$ we can build a real vector space indexed by $A$, which is given by $\mathbb{R}^A := \{a_1 \cdot k_1 + \ldots + a_n \cdot k_n \big| k_i \in \mathbb{R}\}$. This means that a vector $v \in \mathbb{R}^A$ has components $v_a$ indexed by $a \in A$. If the vector $v$ has some other subscripts, like $v_i$, we indicate its component

using parenteses as follows: $(v_i)_a$, $a \in A$. Consider now $A' \subset A$. Given $v \in \mathbb{R}^A$ we can "restrict" it to $A'$, by selecting the components of $v$ indexed by elements in $A'$: $v_{|A'} = v_a$, $a \in A'$.

Lastly, for the results in the following Sections, we do not report the proofs in Section 4.8, but keep them in line, since we think that they help understanding the meaning and the role of all the new definitions which are introduced.

### 4.6.1 Category of Edges and Interval Partitions

We consider the two following categories: Part($[0,1]$) and $\mathcal{E}$. The first one is simply given by the finite partitions of $[0,1]$, that is finite ordered sets of points $0 = v_1 < v_2 < \ldots < v_{n+1} = 1$. This category is a poset with $A < B$ if $A \subset B$. In this category the coproduct is given by the union of two partitions. The category $\mathcal{E}$ instead, is the category obtained by considering the set of edges of finite tree-structures $T$, with the partial order induced by $E_T \simeq V_T - \{r_T\}$. These sets are exactly the finite posets $A$ such that for every $a \in A$, the set $\{a' \in A | a' > a\}$ is a totally ordered set. The arrows in this category are monotone increasing, injective functions between posets. Being a poset, $A \in \mathcal{E}$ can be seen as a category itself. Inside $\mathcal{E}$ we consider the subcategory $\mathcal{E}_2$ made by the set of edges of tree structures without order two vertices, that is $A \in \mathcal{E}$ such that, for all $a \in A$, the set $\max\{a' \in A | a' < a\}$ cannot be of cardinality 1. It is a full subcategory of $\mathcal{E}$.

### 4.6.2 Merge Trees as Functors

Given any tree structure $T$ we can represent it with a functor $F_T$ from the edge set $E_{T_2}$, considered as a poset, to the category Part($[0,1]$) plus a function $w_T : E_{T_2} \to \mathbb{R}_{>0}$, which is just a function from $E_{T_2}$ to $\mathbb{R}_{>0}$, with no added properties. In fact, if we consider a merge tree $T$, every edge $e \in E_{T_2}$ is split in a certain set of edges by the path that goes from $T_2$ to $T$, which means that the interval $[0, w_{T_2}(e)]$ is partitioned by $T$. Up to a scale factor, this means choosing a finite partition of $[0,1]$.

So, having fixed a function $w_T : E_{T_2} \to \mathbb{R}_{>0}$, any functor $F : E_{T_2} \to$ Part($[0,1]$) identifies a merge tree equal up to order 2 vertices to $T_2$: for each $e \in E_{T_2}$, the functor uniquely identifies the merge tree which splits $[0, w_{T_2}(e)]$ using the partition $F(e) \cdot w_{T_2}(e) = \{a_1 w_{T_2}(e) = 0, a_2 w_{T_2}(e), \ldots, a_{n+1} w_{T_2}(e) = w_{T_2}(e)\}$. Given a tree structure $T$ call $F_T$ the associated functor. We can build an association in the other directions for any $F : E \to$ Part($[0,1]$) with $E \in \mathcal{E}$. Given $w : E \to \mathbb{R}_{>0}$ we can first identify the unique tree structure (up to renaming of the vertices) identified by $E$, set $w$ as the weight function and then split the edges according to $F$. We call such merge tree $T_F$, omitting the reference to the weight function when there are no possible ambiguities.

We spend some time to better formalize the space of functors, which is going to be fundamental for the next sections. We set $\mathbf{D} := \{(F : E \to \text{Part}([0,1]), w : E \to \mathbb{R}_{>0}) \big| E \in \mathcal{E}\}$, with arrows induced by arrows in $\mathcal{E}$:

$$f : (F : E \to \text{Part}([0,1]), w) \to (G : E' \to \text{Part}([0,1]), w')$$

is an arrow $f : E \to E'$ in $\mathcal{E}$ such that $F(e) \subset G(f(e))$ and $w(e) = w'(e)$ for all $e \in E$. If $F(e) = G(f(e))$ then we indicate the arrow with $f : F \rightarrowtail G$. Composition follows naturally by composing arrows between the sets of edges in $\mathcal{E}$.

The subcategory $\mathbf{D}_2 := \{(F : E \to \text{Part}([0,1]), w : E \to \mathbb{R}_{>0}) \big| E \in \mathcal{E}_2\}$ is a full subcategory of $\mathbf{D}$.

Now we can formalize the aforementioned correspondences. We define $\mathcal{F}(T) = (F_T, w_T) \in \mathbf{D}_2$ which transforms a merge tree into the couple functor-weight function; viceversa $\mathcal{S}((F, w_F)) = T_F$ takes a couple $(F, w_F) \in \mathbf{D}$ and builds the associated merge tree $T_F$. Moreover let $U_A : A \to \text{Part}([0,1])$, being the canonical tree structure on $A \in \mathcal{E}$, that is the functor such that $U_A(a) = \{0,1\}$ for every $a \in A$. Clearly $\mathcal{F}(T_2) = (U_{E_{T_2}}, w_{T_2})$ and $\mathcal{S}(\mathcal{F}(T_2)) \simeq T_2$. With this notation we can define $\mathcal{C}(T, w_T) = (U_{E_T}, w_T)$. Note that if $T = \mathcal{S}(U_A)$ then $E_T \simeq A$.

We resorted to functors because they allow to easily switch between two kind of representations of one merge tree $T$: $\mathcal{C}(T)$ is in some sense a canonical representation because each edge is sent to $\{0,1\}$ and so there is no added information to the edge set, from the functor; $\mathcal{F}(T)$ instead is a representation of $T$ in terms of $T_2$: it tells how do we have to split each edge in $T_2$ to obtain $T$. Note that $[T] \in \mathcal{T}$ is in bijection with $\{F : E_{T_2} \to \text{Part}([0,1])\} \times \{w_{T_2}\}$.

We have the following isomorphisms which just depend on the vertex set chosen when passing from functors to merge trees: $\mathcal{SF}(T) = T$, $\mathcal{SC}(T) \simeq T$. Moreover, if $T = T_2$, $\mathcal{F}(T) = \mathcal{C}(T)$.

### 4.6.3 Functors parametrizing directions

When referring to elements of $\mathbf{D}$ or $\mathbf{D}_2$ we sometimes omit the weight function, referring just to the functor, for notational convenience. For instance we refer to $F \in \mathbf{D}$ and $\mathcal{S}(F)$ instead of $(F, w_F) \in \mathbf{D}$ and $\mathcal{S}((F, w_F))$. A weight function named $w_F$ is always going to be paired with a functor $F$ such that $(F, w_F) \in \mathbf{D}$.

Given $(H : A \to \text{Part}([0,1]), w_H : A \to \mathbb{R}_{>0}) \in \mathbf{D}$ we call:

$$\mathbf{D}^H = \{(F, f) \big| F : A' \to \text{Part}([0,1]) \in \mathbf{D}, f : A \xrightarrow{\simeq} A' \text{ inducing } H \to F\}$$

The set $\mathbf{D}^H$ is given by all the functors which are defined on the same edge set as $H$, up to isomorphism, and which contain the partitions given by $H$. In fact $f : A \xrightarrow{\simeq} A'$ induces $H \to F$ if $w_H(a) = w_F(f(a))$ (and thus $\mathcal{S}(H)_2 \simeq \mathcal{S}(F)$) and $H(a) \subset F(f(a))$. Note that, if $F, G \in \mathbf{D}^H$, then $\mathcal{S}(F)$ is equal to $\mathcal{S}(G)$ in $\mathcal{T}$. In particular if $H = \mathcal{F}(T_2)$ then $\mathbf{D}^H$ contains $[T]$, represented via $\mathcal{F}(T')$, $T' \in [T]$. Note that each $\mathcal{F}(T')$ can appear multiple times depending on the number of isomorphisms between the set of edges. Moreover, consider $(F, w) \in \mathbf{D}$ with $\mathcal{S}(F) = T$; given $H = \mathcal{F}(T_2)$, then there is $(H', f) \in \mathbf{D}^H$ such that $\mathcal{S}(H') = T$. $H'$ is unique up to isomorphism, while there can be multiple $f$.

As a consequence we have the following Lemma.

**Lemma 6.** *Given $F = \mathcal{C}(T), F' = \mathcal{C}(T')$, if $\mathbf{D}^F \bigcap \mathbf{D}^{F'} \neq \varnothing$ then $F \simeq F'$ in $\mathbf{D}$ and so $\mathbf{D}^F = \mathbf{D}^{F'}$.*

*Proof.* Suppose $F'' : E \to \text{Part}([0,1]) \in \mathbf{D}^F \bigcap \mathbf{D}^{F'}$. Then $E \simeq E_T \simeq E_{T'}$. But since $F(e) = \{0,1\}$ and $F'(e') = \{0,1\}$ for $e \in E_T$ and $e' \in E_{T'}$ respectively, then $F \simeq F'$. □

We have an arrow in $\mathbf{D}^H$ between $(F, f)$ and $(F', f')$ when $f' \circ f^{-1} : A' \to A''$ is an arrow in $\mathbf{D}$. Given $(F, w, f) \in \mathbf{D}^H$, we can build $(F', w', id)$ such that $F'(a) = F(f(a))$ for all $a \in A$ and $w'(a) = w(f(a)) = w_H(a)$. Clearly $(F, f) \simeq (F', id)$. Thus often we can give definitions using $(F', w_H, id)$ and extend them naturally to $(F, w, f) \simeq (F', w_H, id)$. Given $(F, w_H, id)$ and $(G, w_H, id)$ in $\mathbf{D}^H$ we can define $(F \oplus G : A \to \text{Part}([0,1]), w_H, id) \in \mathbf{D}^H$ with the rule $F \oplus G(e) = F(e) \cup G(e)$. Given $F \in \mathbf{D}^H$, $H : A \to \text{Part}([0,1])$, a refinement of $F$ is any functor $F' \in \mathbf{D}^H$ such that we have $F \to F'$, or in other words, up to isomorphism, $F(a) \subset F'(a)$ for every $a \in A$. We write $F < F'$. This defines a partial order relationship. If $F, G$ in $\mathbf{D}^H$, then $F, G < F \oplus G$ and $U_A < F, G$.

Now we can define

$$\mathbf{D}_H = \{(F, f : \mathcal{CS}(H') \rightarrowtail \mathcal{FS}(F)) \big| F \in \mathbf{D} \text{ and } H' \in \mathbf{D}^{\mathcal{FS}(H)})\}$$

We try to make as explicit as possible the meaning of this definition.
First it is clear that $\mathbf{D}_H = \mathbf{D}_{\mathcal{FS}(H)}$. Thus we can assume $H \simeq \mathcal{FS}(H)$.
Consider $(F : E' \to \text{Part}([0,1]), w_F) \in \mathbf{D}$ and $(H' : A' \to \text{Part}([0,1]), w_H, id) \in \mathbf{D}^H$.
Let $\mathcal{FS}(F) : E \to \text{Part}([0,1])$ and $\mathcal{CS}(H') : A \to \text{Part}([0,1])$.
An arrow $f : \mathcal{CS}(H') \rightarrowtail \mathcal{FS}(F)$ is an arrow $f : A \to E$ in $\mathcal{E}$ such that $\mathcal{CS}(H')(a) = \mathcal{FS}(F)(f(a)) = \{0,1\}$ and $w_{\mathcal{CS}(H')}(a) = w_{\mathcal{FS}(F)}(f(a))$ for every $a \in A$. In other words, upon choosing the most convenient functor representation, we are taking a merge tree $\mathcal{S}(H)$, splitting some of its edges obtaining $\mathcal{S}(H')$, and these edges can be embedded into the merge tree $\mathcal{S}(F)$.
Note that, if $H \simeq \mathcal{CS}(H)$, then $\mathbf{D}^H \subset \mathbf{D}_H$. In particular this holds for $H = \mathcal{F}(T_2)$.

**Proposition 13.** *An arrow $f : \mathcal{CS}(H) \rightarrowtail \mathcal{FS}(F)$ induces a unique arrow $\mathcal{CS}(f) : \mathcal{CS}(H) \rightarrowtail F$.*

*Proof.* We have $\mathcal{CS}(H)(e) = \mathcal{FS}(F)(f(e)) = \{0,1\}$ for any edge $e$ in the edge set of $\mathcal{CS}(H)$. Since the edge set of $\mathcal{FS}(F)$ is given by $\mathcal{S}(F)_2$ and those edges are not split by $\mathcal{FS}(F)$ Then we can find those edges also in the edge set of $F$, and since $\mathcal{SFS}(F) \simeq \mathcal{S}(F)$, those edges are not split by $F$.

Thus we have an arrow $\mathcal{CS}(f) : \mathcal{CS}(H) \rightarrowtail F$.

□

We close this section by defining arrows in $\mathbf{D}_H$ as arrows between functors inside $\mathbf{D}^G$, for some functor $G$.

### 4.6.4 Pre-Tangent space and pre-exponential map

With a slight abuse of notation, we call $\mathbf{D}_T$ the set of functors $\mathbf{D}_{\mathcal{F}(T_2)}$ and similarly we used $\mathbf{D}^T$ for $\mathbf{D}^{\mathcal{F}(T_2)}$. With these pieces of notation we can make a first step towards the definition of a tangent space.

**Definition 21.** *Given $[T] \in \mathcal{T}$, the pre-tangent space at $[T]$ is the following set:*

$$\textit{pre-}\mathbf{Tan}_{[T]}(\mathcal{T}) := \{(v, F) \big| F \in \mathbf{D}_T, v \in \mathbb{R}^{\mathcal{CS}(H)}\}$$

*where $F \in \mathbf{D}_T$ stands for the triplet*

$$(F : E \to \textit{Part}([0,1]), w_F : E \to \mathbb{R}_{>0}, f : \mathcal{CS}(H) \rightarrowtail \mathcal{FS}(F))$$

*with $\mathcal{CS}(H) : E' \to \mathbb{R}_{>0}$, $H \in \mathbf{D}^T$, and $\mathbb{R}^{\mathcal{CS}(H)}$ stands for $\mathbb{R}^{E'}$.*

**Proposition 14.** *Take $(v, F) \in \textit{pre-}\mathbf{Tan}_{[T]}(\mathcal{T})$. That is, an element:*

$$(v, F : E \to \textit{Part}([0,1]), w_F : E \to \mathbb{R}_{>0}, f : \mathcal{CS}(H) \rightarrowtail \mathcal{FS}(F))$$

*with $H \in \mathbf{D}^T$.*

*Consider $\mathcal{CS}(f) : \mathcal{CS}(H) \rightarrowtail F$.*

*If $v_e + w_F(e) \leqslant 0$ for every $e \in Im(\mathcal{CS}(f))$ then we can induce a sequence of edits on $\widehat{T} = \mathcal{S}(H) \in [T]$.*

*Proof.* Let $T' = \mathcal{S}(F)$.

By definition, for any $e \in E_{\widehat{T}}$ we have $\mathcal{CS}(f)(e) \in E_{T'}$ with $w_{\widehat{T}}(e) = w_{T'}(e)$. If $w_{T'}(e) + v_e \geqslant 0$ then we can induce the following edits on $T'$, obtaining a merge tree $T''$.
Starting from $T'$ we can edit the edges $e \in E_{\widehat{T}}$ via well defined shrinkings $w_{T''}(e) = w_{\widehat{T}}(e) + v_e$ (which can possibly result in $w_{T''}(e) = 0$). Since $\mathcal{CS}(f) : E_{\widehat{T}} \to E_{T'}$ is injective and monotone increasing, we can insert in $\widehat{T}$ the other edges of $T'$, deleting in the end all the edges with zero valued weights to obtain a valid merge tree $T''$. □

With Proposition 14 we have a consistent way to go from the pre-tangent space, to the space of merge trees. Thus we give the following definitions.

**Definition 22.** *Given $[T] \in \mathcal{T}$, define:*

$$\widetilde{\mathcal{U}}_{[T]} := \{(v, F, \mathcal{CS}(H) \rightarrowtail \mathcal{FS}(F)) \in \textit{pre-}\mathbf{Tan}_{[T]}(\mathcal{T}) \big| w_{\mathcal{S}(H)}(e) + v_e \geqslant 0 \textit{ for } e \in E_{\mathcal{S}(H)}\}$$

*where $E_{\mathcal{S}(H)}$ is the edge set of the merge tree $\mathcal{S}(H)$.*
*The pre-exponential map at $[T]$ is the following function:*

$$\textit{pre-}\mathbf{exp}_{[T]} : \widetilde{\mathcal{U}}_{[T]} \to \mathcal{T}$$

*with pre-$\mathbf{exp}_{[T]}((v, F))$ defined as in Proposition 14.*

**Proposition 15.** *Consider $[T] \in \mathcal{T}$. For every $[T'] \in \mathcal{T}$ there is $(v, F) \in \widetilde{\mathcal{U}}_{[T]}$ such that pre-$\mathbf{exp}_{[T]}((v, F)) \in [T']$.*

*Proof.* Consider the merge tree $T'' = \overbrace{T_2' T_2}$. Then $H = \mathcal{F}(T_2)$ is such that $H \rightarrowtail \mathcal{C}(T'')$. The vector $v \in \mathbb{R}^H$ is given with components $v_e = -w_H(e)$ for every edge. Then the edit path induced by $(v, \mathcal{C}(T''))$ is the deletion of all edges of $T_2$ and the insertion of all edges of $T_2'$. □

**Remark 18.** *The edit paths induced by the elements $(v, F) \in \widetilde{\mathcal{U}}_{[T]}$ on the merge trees in $[T]$ coincide with the set of edit paths which start from some $T' \in [T]$ such that all the edits can be applied simultaneously on $T'$.*

### 4.6.5 Splitting Sets

In our particular situation, we are defining tangent vectors for a space which is a quotient space and thus at every point we have multiple representations of the same object. In the previous sections we provided a first definition of tangent vectors which is strictly bound to the chosen representation of a merge tree, through the choice of a functor $F$. We want to treat some of these representations as equivalent and thus we also need a proper way to transport the vectors $v$ between those representations.

**Definition 23.** *The splitting set in $\mathbb{R}^m$ of a vector $v \in \mathbb{R}^n$ is the set $\nabla_n m(v) = \{v' \in \mathbb{R}^m \big| \sum_i v_i = \sum_j v'_j\}$.*

Note that $\nabla_n m(v) + \nabla_n m(w) = \nabla_n m(v + w)$.

**Definition 24.** *Consider $F < F'$ in some $\mathbf{D}^T$. Let $A = E_{\mathcal{S}(\mathcal{F})}$ and $B = E_{\mathcal{S}(\mathcal{F}')}$. By construction $A \subset B$. The unique edit path made of ghostings between $\mathcal{S}(F')$ and $\mathcal{S}(F)$ (unique up to reordering the edits and up to isomorphisms of merge trees) induces a unique correspondence between $B$ and $A$. For every $a \in A$ there are a sequence of edges $b_1, \dots, b_n$ in $B$ which get merged to the edge $a$ along the edit path which turns $F'$ in $F$. Thus we set $f : B \to A$ such that $f(b_i) = a$. Call $B^a := f^{-1}(a)$. Then we define a multivalued function $\nabla_F F' : \mathbb{R}^F \to \mathbb{R}^{F'}$ such that:*

$$\nabla_F F'(v) = \{v' \in \mathbb{R}^{F'} \big| v'_{|B^a} \in \nabla_1 \# B^a \text{ for all } a \in A\}$$

*Similarly, define the following linear function $\rho^F_{F'} : \mathbb{R}^B \to \mathbb{R}^A$:*

$$(\rho_{F'} F(v))_a = \sum_{b \in B^a} v_b$$

The maps $\rho^F_{F'}$ is a left inverse to $\nabla_F F'$ as proven in the following lemma.

**Lemma 7.** *Consider $F < F'$ in some $\mathbf{D}^T$. Let $A = E_{\mathcal{S}(\mathcal{F})}$ and $B = E_{\mathcal{S}(\mathcal{F}')}$ and $f : B \to A$ the unique correspondence induced as before. Then $\rho^F_{F'} \circ \nabla_F F'(v) = v$.*

*Proof.* For each $a \in A$ by definition we have:

$$v_a \mapsto (v'_b)_{b \in B^a} \mapsto \sum_{b \in B^a} (v'_b) = v_a$$

for any $v' \in \nabla_F F'(v)$. □

The following proposition suggests the idea that $\rho^F_{F'}$ and $\nabla_F F'$ are attempt to transport weight functions inside a class of tree structures $[T]$, trying not to lose information. The proof is straightforward.

**Proposition 16.** *Consider $F < F'$ in some $\mathbf{D}^T$. Let $A = E_{\mathcal{S}(\mathcal{F})}$ and $B = E_{\mathcal{S}(\mathcal{F}')}$ and $f : B \to A$ the unique correspondence induced as before. Let $v_F \in \mathbb{R}^F$ defined as $(v_F)_a = w_F(a)$ and similarly $v_{F'} \in \mathbb{R}^{F'}$ defined as $(v_{F'})_b = w_{F'}(b)$. Then $v_F = \rho^F_{F'}(v_{F'})$ and $v_{F'} \in \nabla_F F'(v_F)$.*

Finally, we prove that along with transporting vectors, we can also transport perturbations of vectors preserving the sign of the single components.

**Lemma 8.** *Consider $\{v_i\}_{i=1}^n$ and $\{w_j\}_{j=1}^m$ sequences of positive numbers, with $\sum v_i = \sum w_j$. Then, given $\{\widehat{v}_i\}_{i=1}^n$ such that $\widehat{v}_i \leqslant v_i$, there is a sequence $\{\widehat{w}_j\}_{i=1}^m$ such that $\widehat{w}_j \leqslant w_j$ and $\sum \widehat{v}_i = \sum \widehat{w}_j$*

*Proof.* If $\sum \widehat{v}_i < 0$ then we can take $\widehat{w}_1 = \sum \widehat{v}_i$ and $\widehat{w}_j = 0$ for all other $j$.

Suppose then $\sum \widehat{v}_i > 0$. Since $\sum \widehat{v}_i \leqslant \sum v_i$, there is $K \in \mathbb{N}$, $K > 0$, such that $\sum_{j=1}^K w_j \leqslant \sum \widehat{v}_i$ and $\sum_{j=1}^{K+1} w_j > \sum \widehat{v}_i$ with the extreme case of $K = m$ and $\sum \widehat{v}_i = \sum v_i$. Then we can take $\widehat{w}_j = w_j$ for all $j = 1, \dots, K$ and $\widehat{w}_{K+1} = \sum v_i - \sum \widehat{v}_i$. □

### 4.6.6 Tangent space and exponential

Now we define a tangent space with the exponential map, by quotienting the pre-tangent space identifying sets of equivalent directions.

**Definition 25.**

$$\mathbf{Tan}_{[T]}(\mathcal{T}) = \textit{pre-}\mathbf{Tan}_{[T]}(\mathcal{T})/\sim$$

*where $(v, F, f : \mathcal{CS}(H) \rightarrowtail \mathcal{FS}(F)) \sim (v', F', f' : \mathcal{CS}(H') \rightarrowtail \mathcal{FS}(F'))$ if*

- *$\mathcal{FS}(F)$ and $\mathcal{FS}(F') \in \mathbf{D}_{T'}$ for some $T'$*
- *there is an isomorphism $g : \mathcal{S}(F)_2 \xrightarrow{\simeq} \mathcal{S}(F')_2$ which induces by restriction an isomorphism $g : \mathcal{S}(H) \xrightarrow{\simeq} \mathcal{S}(H')$;*
- *the map $g$ which gives $\mathcal{S}(H) \simeq \mathcal{S}(H')$, by changing the name of the edges, induces a map $g' : \mathbb{R}^{\mathcal{CS}(H)} \to \mathbb{R}^{\mathcal{CS}(H')}$ such that $g'(v) = v'$.*

The fact that $\sim$ is an equivalence relationship is clarified by the following proposition. Here we just point that inside elements $(v, F, \mathcal{CS}(H) \rightarrowtail \mathcal{FS}(F))$ in $\nu \in \mathbf{Tan}_{[T]}(\mathcal{T})$, we have "fixed" - up to isomorphisms - a representation of $H \in \mathbf{D}_T$, and thus the vectors $v$ are all of the same dimension.

**Proposition 17.** *For each $\nu \in \mathbf{Tan}_{[T]}(\mathcal{T})$, up to isomorphism, there exists $(v, F, \mathcal{CS}(H) \rightarrowtail \mathcal{FS}(F)) \in \nu$ with $F = \mathcal{C}(T_2')$ for some $T_2'$, such that for every $(v', \mathcal{CS}(H') \rightarrowtail \mathcal{FS}(F')) \in \nu$, $F < \mathcal{FS}(F')$. We call such $(v, F)$ a* canonical representation *of $\nu$.*

*Proof.* Since $\mathbf{D}^{T'} \bigcap \mathbf{D}^{T''} = \varnothing$ then there is $T'$ such that for all $(v', \mathcal{CS}(H') \rightarrowtail \mathcal{FS}(F')) \in \nu$, $\mathcal{FS}(F') \in \mathbf{D}_{T'}$.

Let $F := \mathcal{C}(T_2') \simeq \mathcal{F}(T_2')$ and consider $(v', f : \mathcal{CS}(H') \rightarrowtail \mathcal{FS}(F')) \in \nu$. We know that for any edge $e \in E_{\mathcal{S}(H)}$, $\mathcal{FS}(F')(e) = \{0, 1\}$. Since the edge set of $\mathcal{FS}(F')$ is exactly $E_{T_2'}$, $f : E_{\mathcal{S}(H)} \rightarrow E_{T_2'}$ (which gives $\mathcal{CS}(H') \rightarrowtail \mathcal{FS}(F')$) induces $f' : \mathcal{CS}(H') \rightarrowtail F$. Recall that $\mathcal{FS}(F) \simeq F$.

In this way $\mathcal{S}(H')$ identifies the exact same subtree (with the very same vertices) in $\mathcal{S}(F)_2$ and $\mathcal{S}(F')_2$.

For any other $(v'', f'' : \mathcal{CS}(H'') \rightarrowtail \mathcal{FS}(F'')) \in \nu$ we have that the isomorphism $\mathcal{S}(H') \simeq \mathcal{S}(H'')$ induces an isomorphism $\mathcal{CS}(H') \simeq \mathcal{CS}(H'')$. Thus if we build $f''' : \mathcal{CS}(H'') \rightarrowtail F$ we obtain isomorphic elements in $\mathbf{D}_{T'}$.

So $(v', F, f')$ is a canonical representation of $\nu$. □

The following proposition explains how the pre-exponential map behaves on the equivalence classes just defined.

**Proposition 18.** *The pre-exponential map is well defined on $\nu \in \mathcal{U}_{[T]} = \widetilde{\mathcal{U}}_{[T]}/\sim$.*

*Proof.* Consider $(v, f : \mathcal{CS}(H) \rightarrowtail \mathcal{FS}(F))$ and $(v', f' : \mathcal{CS}(H') \rightarrowtail \mathcal{FS}(F'))$ in $\nu$. First we know that $\mathcal{S}(H) \simeq \mathcal{S}(H')$ and thusup to isomorphisms, we can consider $H' = H$. In this way the edit paths to obtain pre-$\mathbf{exp}_T((v, F))$ and pre-$\mathbf{exp}_T((v', F'))$, can be considered starting from the same $\widehat{T} = \mathcal{S}(H) = \mathcal{S}(H')$. We also have $v = v'$. Those edit paths can be split in two parts: one in which we apply on $\widehat{T}$ the edits induced by $v$, and the other in which there are all the insertions determined by $F$ and $F'$. We can clearly start from the second part and obtain $\mathcal{S}(F)$ and $\mathcal{S}(F')$, which, by hypothesis are equal up to order two vertices. Then, inside those trees, we edit the edges corresponding to $\mathcal{S}(H)$ with shrinking induced by the same vector $v$. Since $\mathcal{CS}(H) \rightarrowtail \mathcal{C}(\mathcal{S}(F)_2)$, ghosting and splittings to turn $\mathcal{S}(F)$ into $\mathcal{S}(F)_2$ (and $\mathcal{S}(F')$ into $\mathcal{S}(F')_2$) do not impact $\mathcal{S}(H)$. Thus the final trees are equal up to order two vertices. □

Now we can give the following definition.

**Definition 26.** *Since pre-$\mathbf{exp}_{[T]}$ is well defined on equivalence classes of $\widetilde{\mathcal{U}}_{[T]}/\sim$, we can define:*

$$\mathbf{exp}_{[T]}(\nu) := \textit{pre-}\mathbf{exp}_{[T]}((v, F))$$

*with $\nu \in \mathcal{U}_{[T]}$ and $(v, F)$ canonical representation of $\nu$.*

Given the tangent structure and the exponential map, we can define a length notion for tangent vectors.

**Definition 27.** *Given* $\nu \in \mathbf{Tan}_{[T]}(\mathcal{T})$ *we define* $||\nu|| = inf_{(v,F,f:\mathcal{CS}(H)\rightarrowtail\mathcal{CS}(F))\in\nu}||v||_1 + ||\mathcal{S}(F)|| - ||\mathcal{S}(H)||$.

Note that for any $H \in \mathbf{D}^T$, since $\mathcal{S}(H) \in [T]$, then $||\mathcal{S}(H)|| = ||T||$. The same clearly holds also for $F$, $F' \in \mathbf{D}^{T'}$. Moreover $\mathcal{CS}(H) \rightarrowtail \mathcal{FS}(F)$ implies $||\mathcal{S}(H)|| \leqslant ||\mathcal{S}(F)||$. Thus, $||v||_1 + ||\mathcal{S}(F)|| - ||\mathcal{S}(H)||$ is actually constant on the representatives of $\nu$, and we immediately obtain the following result.

**Lemma 9.** *For each* $\nu \in \mathbf{Tan}_{[T]}(\mathcal{T})$*, let* $(v, F, f)$ *be any representative of* $\nu$*. Then* $||\nu|| = ||v||_1 + ||\mathcal{S}(F)|| - ||T||$.

We start investigating the continuity properties of the exponential map with respect to this notion of length in the tangent space.

**Proposition 19.** *For every* $\nu \in \mathcal{U}_{[T]}$ *we have:* $d_E(T, \mathbf{exp}_T(\nu)) \leqslant ||\nu||$.

*Proof.* Consider $(v, F, f : \mathcal{CS}(H) \rightarrowtail F) \in \nu$. By definition $\mathbf{exp}_T(\nu)$ is obtained via a sequence of edits each with cost equal to the absolute value of the components of $v$ or deletions of edges in $F$ which are not in $\mathcal{S}(H)$. Thus $||\nu||$ is exactly the cost of the edit path induced by $(v, F, f)$ in $\Gamma(T, \mathbf{exp}_T(\nu))$. □

In some special cases the exponential map preserves the length of vectors.

**Proposition 20.** *For every* $\nu \in \mathbf{Tan}_{[T]}(\mathcal{T})$ *such that its canonical representation* $(v, F, f)$ *has* $v_e \geqslant 0$ *for all* $e$*, we have* $d_E(T, \mathbf{exp}_T(\nu)) = ||\nu||$.

*Proof.* The merge tree $T' = \mathbf{exp}_T(\nu)$ is obtained from $T$ with with an edit path. If we take $T_2$ and $(v, F, f)$ the canonical representation of $\nu$. Any edge $e$ in $T_2'$ is either obtained via shrinking by $v_e$ an edge of $T_2$, or by inserting an edge of weight $w_{T_2'}(e)$. Thus:

$$\sum_{e\in E_{T_2'}} w_{T_2'}(e) = \sum_{e\in f(E_{T_2})} w_{T_2}(e) + v_e + \sum_{e\in E_{T_2'} - f(E_{T_2})} w_{T_2'}(e) = \sum_{e\in f(E_{T_2})} w_{T_2}(e) + \sum_{e\in E_{T_2'}} w_{T_2'}(e)$$

Which implies $\big|||T|| - ||T'||\big| = ||\nu|| \leqslant d_E(T, T')$.

□

### 4.6.7 Linear Structure

The tangent space at a point $[T]$ is very complicated and the directions that can be taken starting from $[T]$ are so different that it is not easy to define a linear structure to combine them. We start by considering some subsets of the tangent space, whose direction go *parallel* to $\mathcal{T}^N$, with $N = dim(T) := dim(T_2)$.

**Definition 28.** *For every $[T]$ we can consider the following subset of the tangent space:*

$$\mathbf{Tan}_{[T]}(\mathcal{T})^{\parallel} := \{(v, F, f : \mathcal{CS}(H) \rightarrowtail \mathcal{FS}(F)) \big| H, F \in \mathbf{D}_T\}/\sim$$

*Similarly we call $\mathcal{U}^{\parallel}_{[T]} = \mathcal{U}_{[T]} \bigcap \mathbf{Tan}_{[T]}(\mathcal{T})^{\parallel}$.*

Note that, since two elements $(v, F)$ and $(v', F')$ are equivalent with respect to $\sim$ only if $\mathcal{FS}(F)$ and $\mathcal{FS}(F') \in \mathbf{D}_{T'}$ for some $T'$, then $\mathbf{Tan}_{[T]}(\mathcal{T})^{\parallel} \subset \mathbf{Tan}_{[T]}(\mathcal{T})$. That is, the equivalence classes in $\mathbf{Tan}_{[T]}(\mathcal{T})^{\parallel}$ are compatible with the ones of $\mathbf{Tan}_{[T]}(\mathcal{T})$. Thus $\mathcal{U}^{\parallel}_{[T]}$ is well defined.

Moreover $\mathbf{Tan}_{[T]}(\mathcal{T})^{\parallel}$ can be strongly characterized. Since the edge set of $\mathcal{FS}(F)$ is $E_{T_2}$, this must hold also for $\mathcal{CS}(H)$. In fact if $E$ is the edge set of $\mathcal{CS}(H)$, we have $E_{T_2} \subset E \subset E_{T_2}$. Thus we have $H \simeq F \simeq \mathcal{C}(T_2)$. In turns this implies that, up to isomorphism, $\mathbf{Tan}_{[T]}(\mathcal{T})^{\parallel}$ can be represented by couples $(v, \mathcal{C}(T_2))$ with $v \in \mathbb{R}^{E_{T_2}}$. We indicate this fact with the notation $\mathbb{R}^{E_{T_2}} \approx \mathbf{Tan}_{[T]}(\mathcal{T})^{\parallel}$.

This notation allows a local linear structure to be recovered in the space of merge trees, as we formalize in the following lemma.

**Lemma 10.** *Consider $[T]$ with $T = T_2$, and $v \in U_T \subset \mathbb{R}^T \approx \mathbf{Tan}_{[T]}(\mathcal{T})^{\parallel}$. Let $v_T := (w_{T_2}(e))_{e \in E_{T_2}}$. Then the merge tree $\mathbf{exp}_{[T]}(v)$ can be identified with the vector $v_T + v$.*

*Consequently we have $\mathbf{exp}_{[T]}(U_T) \approx U_T + v_T = \mathbb{R}^T_{\geqslant 0}$, where $\mathbb{R}^T_{\geqslant 0}$ is the set of vectors $(v_e)_{e \in E_T}$ such that $v_e \geqslant 0$.*

*Proof.* Starting from $T$, we obtain $T + v = \mathbf{exp}_{[T]}(v)$ by editing every edge $e \in E_T$ with $v_e$ such that $w_{T+v}(e) = w_T(e) + v_e$. Therefore the vector $v' = v_T + v$ can be seen as the set of weights on the set of edges $E_T$, with possibly zero valued weights. The merge tree $T + v$ is the unique merge tree up to isomorphism obtained by deleting from $E_T$ the edges with $v'_e = 0$ and with the other weights given by the support of $v'$ (its non zero components). □

However, as we will point out with some other results, this linear structure is not in general compatible with the metric $d_E$: straight lines between trees are not geodesics, in general.

In a similar fashion, we define a linear structure on $\mathbf{Tan}_{[T]}(\mathcal{T})^{\parallel}$. Since there is always the choice of some vertex set involved, for us it is enough to define a linear structure up to isomorphism of functors.

**Definition 29.** *Having fixed a representation of $T_2$, the linear structure on isomorphism classes of $\mathbf{Tan}_{[T]}(\mathcal{T})^{\parallel}$ is given by the one of $\mathbb{R}^{E_{T_2}}$.*

Moreover, since $||T|| = ||H||$ for all $(v, F, f : \mathcal{CS}(H) \rightarrowtail \mathcal{FS}(F))$ with $H, F \in \mathbf{D}_T$, we have $||\nu|| = ||v||_1$, which induces a proper norm on $\mathbf{Tan}_{[T]}(\mathcal{T})^{\parallel}$.

**Proposition 21.** *Consider* $[T] \in \mathcal{T}^N$*, with* $N = dim(T_2)$*, and* $\nu \in \mathcal{U}^{\|}_{[T]}$*. Then* $T' = \mathbf{exp}_{[T]}(\nu) \in \mathcal{T}^N$ *and there is a map, induced by* $\nu$*,* $\rho : \mathbf{Tan}_{[T]}(\mathcal{T}) \to \mathbf{Tan}_{[T']}(\mathcal{T})$*, called* parallel transport map.

*Moreover we can induce a linear map:* $\rho^{\|} : \mathbf{Tan}_{[T]}(\mathcal{T})^{\|} \to \mathbf{Tan}_{[T']}(\mathcal{T})^{\|}$.

*Proof.* Since $\nu \in \mathbf{Tan}_{[T]}(\mathcal{T})^{\|}$ can only shrink or delete edges of $T_2$, clearly $[T'] = \mathbf{exp}_T(\nu) \in \mathcal{T}^N$.

Consider $(v, T_2, id : \mathcal{C}(T_2) \rightarrowtail \mathcal{C}(T_2))$ canonical representation of $\nu$ and $(v', F', f' : \mathcal{CS}(H') \to F')$ canonical representation of $\nu' \in \mathbf{Tan}_{[T]}(\mathcal{T})$, with $\mathcal{C}(T_2) < H'$. We have $g : E_{T'} \to E_{T_2}$ injective map in $\mathcal{E}$ and thus we can consider $H'_{|g(E_{T'})}$. In fact we have:

$$E_{T'} \xrightarrow{g} E_{T_2} \xrightarrow{H'} \text{Part}([0,1])$$

In order to obtain the following map in $\mathbf{D}$:

$$\mathcal{CS}(H'_{|g(E_{T'})}) \rightarrowtail \mathcal{CS}(H') \rightarrowtail F'$$

we need to take care of the weight functions, which must take into account that we moved from $[T]$ to $[T']$. Clearly we have the following injective maps between edge sets:

$$E_{\mathcal{S}(H'_{|g(E_{T'})})} \rightarrowtail E_{\mathcal{S}(H')} \rightarrowtail E_{\mathcal{S}(F')}$$

Call $f : E_{\mathcal{S}(H'_{|g(E_{T'})})} \rightarrowtail E_{\mathcal{S}(F')}$ this composition.

We modify the weight functions using $w_{T'}$. The functor $H'_{|g(E_{T'})}$ is coupled with the function $w_{H''} : g(E_{T'}) \to \mathbb{R}_{>0}$ defined as $w_{H''}(g(e)) = w_{T'}(e)$. Now that we have a proper couple $(H'_{|g(E_{T'})}), w_{H''})$, which we refer to as $H''$, as opposed to $H'_{|g(E_{T'})} = (H'_{|g(E_{T'})}, w_{H'|g(E_{T'})})$, and we can consider the merge tree $\mathcal{S}(H'')$. Note that $\mathcal{C}(T'_2) < H''$.

To change $w_{F'}$ and obtain $w'_{F'}$ we must be very careful. For any $e \in E_{\mathcal{S}(F')} - f'(E_{\mathcal{S}(H')})$, we define $w'_{F'}(e) = w_{F'}(e)$. For any edge in $e \in f(E_{\mathcal{S}(H'_{|g(E_{T'})})})$ we set $w'_{F'}(e) = w_{\mathcal{S}(H'')}(f^{-1}(e))$. The remaining edges are the ones in $f'(E_{\mathcal{S}(H')}) - f(E_{\mathcal{S}(H'_{|g(E_{T'})})})$, which correspond to the edges which are obtained by splitting edges in $E_{T_2}$ which have already been deleted to obtain $T'$. For such edges we set $w'_{F'}(e) = \max\{0, v'_e\}$. Lastly, we call $F'' = (F', w'_{F'})$, which may require some deletions to remove edges $e$ with $w'_{F'}(e) = 0$.

In this way we obtain:

$$f'' : \mathcal{CS}(H'') \rightarrowtail F''$$

arrow in $\mathbf{D}$.

Similarly, from $v' \in \mathbb{R}^{\mathcal{CS}(H')}$ we can obtain by restriction the vector $v'_{|\mathcal{CS}(H'_{|g(E_{T'})})} \in \mathbb{R}^{\mathcal{CS}(H'_{|g(E_{T'})})}$ by selecting the components $v'_e$ such that $e$ lies in the image of $\mathcal{CS}(H'_{|g(E_{T'})}) \rightarrowtail$

$\mathcal{CS}(H')$. Since the edge set of $H''$ and $H'_{|g(E_{T'})}$ is the same we obtain $v'_{|\mathcal{CS}(H'_{|g(E_{T'})})} = v'_{|\mathcal{CS}(H'')} \in \mathbb{R}^{\mathcal{CS}(H'')}$. We call this vector $v''$.
Thus we define $\rho((v', F', f')) := (v'', F'', f'' : \mathcal{CS}(H'') \rightarrowtail F'')$.

The tangent vector $\rho((v', F', f'))$ is a canonical representation of an element in $\mathbf{Tan}_{[T']}(\mathcal{T})$.
Note that $\rho$ in general does not preserve the lenght of tangent vectors.

$$||\rho((v', F', f'))|| = \sum_{e \in Im(f'')} |v'_e| + \sum_{e \in E_{\mathcal{S}(F')} - f'(E_{\mathcal{S}(H')})} w_{F'}(e) + \sum_{e \in f'(E_{\mathcal{S}(H')}) - f(E_{\mathcal{S}(H'_{|g(E_{T'})})})} \max\{0, v'_e\}$$

So any time $v'_e < 0$ for some $e \in f'(E_{\mathcal{S}(H')}) - f(E_{\mathcal{S}(H'_{|g(E_{T'})})})$ the lenght of vectors changes.

Thus if $v'_e > 0$ for all $e \in f'(E_{\mathcal{S}(H')}) - f(E_{\mathcal{S}(H'_{|g(E_{T'})})})$, $||\rho((v', F', f'))|| = ||(v', F', f')||$.

Now we turn to $\rho^{\|}$. Suppose $(v, T_2, id : \mathcal{C}(T_2) \rightarrowtail \mathcal{C}(T_2))$ such that $v \in U_{T_2} \approx \mathcal{U}^{\|}_{[T]}$, and $(v', T_2, id : \mathcal{C}(T_2) \rightarrowtail \mathcal{C}(T_2))$ represented by $v' \in \mathbb{R}^{E_{T_2}} \approx \mathbf{Tan}_{[T]}(\mathcal{T})^{\|}$. Starting from $T_2$, we can edit it obtaining $T' = \mathbf{exp}_{[T]}(v)$ whose set of edges is $E^+_{T_2}$ built as follows. Take $v_T \in \mathbb{R}^{E_{T_2}}$, with components $(v_T)_e = w_{T_2}(e)$. Then $E^+_{T_2}$ is the set $e \in E_{T_2}$ such that $(v_T)_e - v_e > 0$.
Fix a representation $\mathbb{R}^{E_{T'_2}} \approx \mathbf{Tan}_{[T']}(\mathcal{T})^{\|}$. Consider $H = \mathcal{C}(T'_2)$ and $H' = \mathcal{C}(T')$; $H < H'$. If $H'$ is not isomorphic to $H$, then $E^+_{T_2} \notin \mathcal{E}_2$ and, of course, $v'_{|E^+_{T_2}} \notin \mathbb{R}^H$. Thus the solution is to consider $H$, and transport the vector $v'_{|E^+_{T_2}}$ from $\mathbb{R}^{\mathcal{CS}(H')}$ to $\mathbb{R}^H$ with $\rho^H_{H'}$.

Putting all the pieces together the map $\rho^{\|} : \mathbb{R}^{E_{T_2}} \rightarrow \mathbb{R}^{E_{T'_2}}$ is defined as $\rho^{\|}(v') = \rho^H_{H'}(v'_{|E^+_{T_2}})$. This map is linear and, in general, it does not preserve the norm of vectors.

□

**Remark 19.** *The fact that $\rho$ does not preserve the lenght of all tangent vectors, suggests that we could tweak the tangent space definition. By "enlarging" the set of tangent vectors, allowing also "negative" values, we could avoid taking max$\{0, v'_e\}$ (using the notation of the proof) to obtain positive weights, and simply take $v'_e$. This may result in a more natural definition of $\rho$.*

It is clear that $\rho^{\|}$ it is an isomorphism of vector spaces any time $dim(\mathbf{exp}_T(\nu)) = dim(T)$, that is $E^+_{T_2} = E_{T_2}$ (and so $H = H'$), but when we lose for instance one dimension and in some sense we reach the border of $\mathcal{T}^N - \mathcal{T}^{N-1}$, also the parallel tangent space drops one dimension. Thus $\rho^{\|}$ is no more injective. We further investigate the properties of such maps with a series of results.

**Proposition 22.** *Consider $[T] \in \mathcal{T}^N$, with $N = dim(T_2)$, $\nu \in \mathbf{Tan}_{[T]}(\mathcal{T})^{\|}$ and $T' = \mathbf{exp}_T(\nu)$).*

*Then induce the map:* $\rho^{\|} : \mathbf{Tan}_{[T]}(\mathcal{T})^{\|} \to \mathbf{Tan}_{[T']}(\mathcal{T})^{\|}$.

*We can build at least one linear map* $\nabla^{\|} : \mathbf{Tan}_{[T']}(\mathcal{T})^{\|} \to \mathbf{Tan}_{[T]}(\mathcal{T})^{\|}$ *such that* $\rho^{\|} \circ \nabla^{\|} = id$.

*Proof.* As in the previous proposition we fix a representation of $T_2$ along with the correspondence $\mathbb{R}^{E_{T_2}} \approx \mathbf{Tan}_{[T]}(\mathcal{T})^{\|}$. Let $v$ represent $\nu$ and, as before, obtain $T' = \mathbf{exp}_T(v)$ whose set of edges is $E^+_{T_2}$. Thus $E^+_{T_2} \subset E_{T_2}$ and we fix also $\mathbb{R}^{E_{T'_2}} \approx \mathbf{Tan}_{[T']}(\mathcal{T})^{\|}$. Let $H = \mathcal{C}(T'_2)$ and $H' = \mathcal{C}(T')$; $H < H'$. Then $\rho^{\|} : \mathbb{R}^{E_{T_2}} \to \mathbb{R}^{E_{T'_2}}$ is defined as $\rho^{\|}(v') = \rho^H_{H'}(v'_{|E^+_{T_2}})$.

Lets call $i : E^+_{T_2} \hookrightarrow E_{T_2}$ and consider $\nabla_H H' : \mathbb{R}^H \to \mathbb{R}^{H'}$. Given $v \in \mathbb{R}^H$, we take $v' \in \nabla_H H'(v)$ and extend it to $\widehat{v} \in \mathbb{R}^{E_{T_2}}$ by setting $\widehat{v}_{i(e)} = v'_e$ for all $e \in E^+_{T_2}$ and $\widehat{v}_{e'} = 0$ for all $e' \in E_{T_2} - E^+_{T_2}$.

Finally, the map $\nabla^{\|} : \mathbb{R}^{E_{T'_2}} \to \mathbb{R}^{E_{T_2}}$ is defined as $\nabla^{\|}(v) := \widehat{v}$.

This map depends on the choice of $v' \in \nabla_H H'(v)$ and is well defined up to isomorphism classes in $\mathbf{Tan}_{[T']}(\mathcal{T})^{\|}$. For us the choice of $v' \in \nabla_H H'(v)$ is irrelevant, so we indicate with $\nabla^{\|}$ any map obtained with a choice of $v'$ for any $v$.

Lastly, we have: $\rho^{\|} \circ \nabla^{\|}(v) = \rho^{\|}(\widehat{v}) = \rho^H_{H'}(\widehat{v}_{|E^+_{T_2}}) = \rho^H_{H'}(v') = v$.

□

**Remark 20.** *For notational convenience, in what follows we sometimes use the following notation: if* $T' = \mathbf{exp}_{[T]}(\nu)$ *we say* $T' = T + \nu$.

Having established a normed space structure in $\mathbf{Tan}_{[T]}(\mathcal{T})^{\|}$ we can investigate the continuity properties of the exponential map. We have already verified that such map is continuous on "lines going through the origin" of the tangent space, in the sense that $d_E(T, T + \nu) \leqslant ||\nu||$, but now we obtain a stronger continuity result, which only holds for $\mathcal{U}^{\|}_{[T]}$. Before doing so, the following lemma tells us that the parallel transport previously defined is compatible with the linear structure induced by the representation $\mathbb{R}^{E_{T_2}}$.

**Lemma 11.** *Consider* $[\widehat{T}] \in \mathcal{T}$. *Suppose* $\widehat{T} = \widehat{T}_2$ *and fix a representation* $\mathbb{R}^{\widehat{T}} \approx \mathbf{Tan}_{[\widehat{T}]}(\mathcal{T})^{\|}$.

*Let* $T = \mathbf{exp}_{[\widehat{T}]}(\nu)$ *and* $T' = \mathbf{exp}_{[\widehat{T}]}(\nu')$. *Consider* $v$ *and* $v'$ *representing respectively* $\nu$ *and* $\nu'$ *in* $\mathbb{R}^{\widehat{T}}$. *Then* $T + \rho(v' - v) = T'$. *In particular* $T + \rho(-v) = \widehat{T}$.

*Proof.* Let $v_{\widehat{T}} \in \mathbb{R}^{\widehat{T}}$ be the vector with $(v_{\widehat{T}})_e = w_{\widehat{T}}(e)$. The merge tree $T$ can be identified with $v_{\widehat{T}} + v$. We call $E^+_{T_2}$ its support (components with positive values).

Consider the vector $v' - v$. Since $v', v \in U_{T_2}$, then $v_e, v'_e \geqslant w_{\widehat{T}}(e)$ for all $e \in E_{T_2}$. Thus for any $e \notin E^+_{T_2}$ we have $v_e = -w_{\widehat{T}}(e)$, and so $v'_e - v_e \geqslant 0$. In other words the parallel transport of $v' - v$ from the tangent in $[\widehat{T}]$ to the tangent in $[T]$, leaves $v' - v$

unchanged, since the components $(v - v')_{|E^+_{T_2}}$ are left as they are, but since $v'_e - v_e \geqslant 0$ for $e \notin E^+_{T_2}$ also the others are unchanged.

Let $v_{\widehat{T}}$ be the vector in $\mathbb{R}^{\widehat{T}}$ such that $(v_{\widehat{T}})_e = w_{\widehat{T}}(e)$. Then $T$ can be identified with the vector $v_{\widehat{T}} + v$. Thus $T + \rho(v' - v)$ becomes the vector $v_{\widehat{T}} + v + v' - v$ which identifies $T'$. The case $v' = 0$ holds the particular case $\rho(-v)$. □

With the previous lemma we easily obtain the continuity of the exponential map when restricted to $U_{T_2}$.

**Proposition 23.** *Consider $[\widehat{T}] \in \mathcal{T}$. Suppose $\widehat{T} = \widehat{T}_2$ and fix a representation $\mathbb{R}^{\widehat{T}} \approx \mathbf{Tan}_{[\widehat{T}]}(\mathcal{T})^{\|}$.*

*Let $T = \mathbf{exp}_{[\widehat{T}]}(\nu)$ and $T' = \mathbf{exp}_{[\widehat{T}]}(\nu')$. Consider $v$ and $v'$ representing respectively $\nu$ and $\nu'$ in $\mathbb{R}^{\widehat{T}}$, then $d_E(T, T') \leqslant ||v - v'||$.*

*Proof.* We apply Lemma 11 and since $\rho$ preserves the lenght of $v' - v$, we conclude with Proposition 19. □

### 4.6.8 Geodesics Decomposition

In this section we develop some decomposition properties for geodesics in the merge trees space, parametrized by mappings. The ideal situation would be to retrieve some sort of Pythagora's Theorem to decompose "variance" between merge trees. Since our metric space is much closer to the Manhattan norm $|| \cdot ||_1$, compared to the one induced by the standard scalar product in $\mathbb{R}^n$, we do not expect an equally strong result, but still we are able to retrieve some useful decomposition properties using tangent spaces.

Before the next proposition we recall some pieces of notation used to introduce mappings (see Chapter 2). Consider $T$, $T' \in \mathcal{T}$ and $M \in Mapp(T, T')$. Then $M$ parametrizes edit paths of the form $T \xrightarrow{\gamma_g \circ \gamma_d} T_M \xrightarrow{\gamma_s^T} T'_M \xrightarrow{(\gamma_g^{T'} \circ \gamma_d^{T'})^{-1}} T'$. Where $\gamma_g \circ \gamma_d$ is a path given by deletions and ghostings on $T$, then with $\gamma_s^T$ we apply some shrinkages on $T$, and lastly we split and insert edges with $(\gamma_g^{T'} \circ \gamma_d^{T'})^{-1}$ to obtain $T'$.

Now we start establishing relationships between mappings and tangent vectors.

**Definition 30.** *Consider $[T]$, $[T'] \in \mathcal{T}$ and $M \in Mapp(T_2, T'_2)$.*

*A couple of vectors $\nu_T \in \mathbf{Tan}_{[T]}(\mathcal{T})$ and $\nu_{T'} \in \mathbf{Tan}_{[T']}(\mathcal{T})$ such that $\mathbf{exp}_{[T]}(\nu_T) = \mathbf{exp}_{[T']}(\nu_{T'})$ and $d_E(T, T') = ||\nu_T|| + ||\nu_{T'}||$ is called a tangent geodesic decomposition.*

*A tangent geodesic decomposition, $\nu_T$ and $\nu_{T'}$, such that $\nu_T \in \mathbf{Tan}_{[T]}(\mathcal{T})^{\|}$ and $\nu_{T'} \in \mathbf{Tan}_{[\mathbf{exp}_T(\nu_T)]}(\mathcal{T})^{\|}$ is called parallel tangent geodesic decomposition.*

*Moreover, if the edits induced by $\nu_T$ and $\nu_{T'}$ are contained in a minimizing mapping $M \in Mapp(T, T')$ then those same vectors form a (parallel) tangent decomposition of $M$.*

**Proposition 24.** *Given $\nu_T$, $\nu_{T'}$ tangent geodesic decomposition, we have $d_E(T, T + \nu_T) = ||\nu_T||$ and $d_E(T', T' + \nu_{T'}) = ||\nu_{T'}||$.*

*Proof.* We know from previous results that $d_E(T, T + \nu_T) \leqslant ||\nu_T||$ and $d_E(T', T' + \nu_{T'}) \leqslant ||\nu_{T'}||$. Moreover, by hypothesis we have $T + \nu_T = T' + \nu_{T'}$. Suppose $d_E(T, T + \nu_T) < ||\nu_T||$. Then $d_E(T, T') \leqslant d_E(T', T' + \nu_{T'}) + d_E(T, T + \nu_T) < ||\nu_T|| + ||\nu_{T'}|| = d_E(T, T')$, which is a contradiction. □

**Proposition 25.** *Consider $[T]$, $[T'] \in \mathcal{T}$ and $M \in Mapp(T_2, T_2')$.*

*Then we can find a parallel tangent geodesic decomposition of $M$, called $\nu_T^{\parallel}$ and $\nu_{T'}^{\parallel}$, such that $\nu_T^{\parallel} \in \mathbf{Tan}_{[T]}(\mathcal{T})^{\parallel}$ and $\nu_{T'}^{\parallel} \in \mathbf{Tan}_{[T']}(\mathcal{T})^{\parallel}$, with $T + \nu_T^{\parallel} = T'_M$ and $T' + \nu_{T'}^{\parallel} = T'_M$.*

*Proof.* By selecting the deletions of the form $(D, v') \in M$, we obtain vector $\nu_{T'}^{\parallel} \in \mathbf{Tan}_{[T']}(\mathcal{T})^{\parallel}$ such that $\mathbf{exp}_{T'}(\nu_{T'}^{\parallel}) = T'_M$. Then we observe that there is a vector $\nu \in \mathbf{Tan}_{[T]}(\mathcal{T})^{\parallel}$, obtained in a similar fashion, such that $\mathbf{exp}_T(\nu) = T_M$. The shrinkings contained in the mapping then induce $\nu' \in \mathbf{Tan}_{[T_M]}(\mathcal{T})^{\parallel}$ such that $\mathbf{exp}_{T_M}(\nu') = T'_M$.
Let $\rho^{\parallel} : \mathbf{Tan}_{[T]}(\mathcal{T})^{\parallel} \to \mathbf{Tan}_{[T_M]}(\mathcal{T})^{\parallel}$ induced by $\nu$.
We can consider a map $\nabla^{\parallel} : \mathbf{Tan}_{[T_M]}(\mathcal{T})^{\parallel} \to \mathbf{Tan}_{[T]}(\mathcal{T})^{\parallel}$ such that $\rho^{\parallel} \circ \nabla^{\parallel} = id$.
We claim that $\nu_T^{\parallel} := \nu + \nabla^{\parallel}(\nu')$ is such that $\mathbf{exp}_T(\nu_T^{\parallel}) = T'_M$. Let $v$ and $v'$ representations in $\mathbb{R}^{T_2}$ of $\nu$ and $\nabla^{\parallel}(\nu')$ respectively. Note that, by construction, the supports of $v$ and $v'$ do not intersect: $v$ only delete edges and $v'$ only shrinks edges in $T_M$.
Then, for any edge $e \in E_{T_2}$ which is deleted by $v$, we have $v_e = -w_{T_2}(e)$. For any other edge we shrink it by $v'_e$. Once we do all the deletions in $v$, we obtain $T_M$. We are left to apply the edits induced by $v'$. However, editing $T_M$ with those edits is equal to editing it with $\rho^{\parallel}(\nabla^{\parallel}(\nu')) = \nu'$, that is $\mathbf{exp}_{T_M}(\nu') = T'_M$.
The result about the costs follows immediately. □

Thus for any couple of trees $[T]$ and $[T']$ in $\mathcal{T}$, we can use a minimizing mapping to find a geodesic which can be decomposed into two parts: one parallel to $[T]$ and the second parallel to $[T']$. These two parts are also geodesics on their own, as pointed out by Proposition 24.

We complement Proposition 24 with the following propositions.

**Proposition 26.** *If $d_E(T, \mathbf{exp}_T(\nu)) = ||\nu||$, then for any $\nu'$ and $\nu''$ such that $\nu = \nu' + \nu''$ and $||\nu|| = ||\nu'|| + ||\nu''||$, we have $d_E(T, \mathbf{exp}_T(\nu')) = ||\nu'||$ and $d_E(\mathbf{exp}_T(\nu), \mathbf{exp}_T(\nu'')) = ||\nu''||$.*

*Proof.* Suppose that $d_E(T, \mathbf{exp}_T(\nu')) < ||\nu'||$. Then $||\nu|| = ||\nu'|| + ||\nu''|| > d_E(T, \mathbf{exp}_T(\nu')) + ||\nu''|| > d_E(T, \mathbf{exp}_T(\nu))$. Absurd. Similarly we prove $d_E(\mathbf{exp}_T(\nu), \mathbf{exp}_T(\nu'')) = ||\nu''||$. □

**Proposition 27.** *If $d_E(T, \mathbf{exp}_T(\nu)) = ||\nu||$ then $\mathbf{exp}_T(\lambda\nu))$ for $\lambda \in [0,1]$ is an always minimizing geodesic.*

*Proof.* Since $\lambda \in [0,1]$, from $\nu = \lambda\nu + (1-\lambda)\nu$ we have $||\nu|| = \lambda||\nu|| + (1-\lambda)||\nu||$ and thus the result follows from the previous proposition. □

We propose one last definition, which is yet to be investigated, but we think that can play an important role when working in tangent spaces since, from the point of view of the tangent point, allows to travel parallel to the tangent point for as much length as possible. Thus, for instance, we can capture as much as possible of the variability between two merge trees staying in the parallel tangent space.

**Definition 31.** *A $T$-maximal (parallel) tangent geodesic decomposition is a (parallel) decomposition $\nu_T^m$ and $\nu_{T'}^m$ such that $||\nu_T^m|| \geqslant ||\nu_T||$ for all other (parallel) decompositions $\nu_T$ and $\nu_{T'}$.*

**Claim 2.** *The decomposition of a minimizing mapping $M$ found as in Proposition 25 is a $T$-maximal parallel geodesic decomposition.*

## 4.7 Frechét Mean Approximation

We conclude this chapter by employing all the machinery defined in the previous sections to build an iterative variational algorithm to approximate a Frechét Mean of a finite set of merge trees. The main idea behind the algorithm is to try to exploit the continuity of the exponential map to solve some optimization problem in a tangent space and then map the results back to the space of merge trees. The followed approach and the final result are not far from the algorithm proposed by Pennec (2006) exploiting results obtained by Karcher (1977).

Consider a set of merge trees $\{[T_i]\}_{i=1}^n$; consider a merge tree $[T] \in \mathcal{T}^N$ and $T = T_2$. For all $i$, to lighten the notation, we call $\nu_i$ and $\nu_i'$ a parallel decomposition of a $M_i$ minimizing mapping between $T$ and $T_i$. Thus we can write $T + \nu_i - \nu_i'$, which is to be interpreted as $(T + \nu_i) + \rho(-\nu_i)$, with $\rho : \mathbf{Tan}_{[T']}(\mathcal{T})^{\|} \to \mathbf{Tan}_{[T'+\exists_i']}(\mathcal{T})^{\|}$. Recall that we have $cost(M_i) = ||\nu_i|| + ||\nu_i'||$. Thus let $\sum_i cost(M_i) = \sum_i ||\nu_i|| + ||\nu_i'||$. If we call $T_i' = T + \nu_i$ we can look for:

$$\widehat{\nu} = \underset{\nu \in \mathcal{U}_{[T]}^{\|}}{\arg\min} \sum_i (d_E(T+\nu, T_i') + ||\nu_i'||)^p \leqslant \sum_i (||\nu_i|| + ||\nu_i'||)^p$$

Define $v_T := (w_T(e))_{e \in E_T}$ and consider $v_i + v_T$ representations of $T_i'$ in $\mathbb{R}^T_{\geqslant 0}$ (and thus $v_i$ representation of $\nu_i \in U_{T_2}$). Then we can consider the following optimization problem:

$$\widehat{v} = \underset{v \in U_T}{\arg\min} \sum_i (||v_T + v - (v_i + v_T)|| + ||\nu_i'||)^p$$

$$\widehat{v} = \arg\min_{v} \sum_{i} (||v - v_i|| + ||\nu_i'||)^p$$

Which is a constrained minimization problem in $\mathbb{R}^{E_{T_2}}$. For instance, with $p = 1$, we obtain:

$$\widehat{v} = \arg\min_{v \in U_T} \sum_{i} ||v - v_i||$$

and thus $\widehat{v}$ is given by the median of $v_i$ on each component.

Recall that, by previous results, $d_E(T + \widehat{v}, T + v_i) \leqslant ||\widehat{v} - v_i||$.

Suppose we obtain a minimizer $\widehat{v}$; the we have a merge tree $\widehat{T} = T + \widehat{v}$ such that:

$$\sum_i (d_E(\widehat{T}, T_i') + d_E(T_i', T_i))^p \leqslant \sum_i (||\widehat{v} - v_i|| + ||\nu_i'||)^p \leqslant \sum_i (||\nu_i|| + ||\nu_i'||)^p = \sum_i d_E(T, T_i)^p$$

and so:

$$\sum_i d_E(\widehat{T}, T_i)^p \leqslant \sum_i d_E(T, T_i)^p$$

Note that if $\sum_i (||\widehat{v} - v_i|| + ||\nu_i^-||)^p < \sum_i (||v_i|| + ||\nu_i'||)^p$ then actually $\sum_i d_E(\widehat{T}, T_i)^p < \sum_i d_E(T, T_i)^p$.

This can be considered a step in an iterative procedure to approximate a Frechét Mean of $\{[T_i]\}$. Suppose we are at the $k-$th iteration. Then starting from $T^k$ what we do is:

1. calculate the mappings $M_i^k = d_E(T_k, T_i)$;
2. calculate the decompositions of $M_i^k$, $\nu_i^k$ and $\nu_i'^k$, using $\nabla^{\|}$ as in Proposition 25;
3. "align the trees " to obtain $v_i^k$, the representations of $\nu_i^k$ in $\mathbb{R}^{T_k}$
4. solve $\widehat{v} = \arg\min_v \sum_i (||v - v_i^k|| + ||\nu_i'^k||)^p$
5. obtain $T_{k+1} = T_k + \widehat{v}$;

Of course if the vectors $\nu_i^k$ in the decomposition of the mappings between $T_k$ and $T_i$ are given by $-\widehat{v} + v_i^{k-1}$ we are in a fixed point of the algorithm.

We close the chapter with a series of claims regarding this approximation procedure, which are going to be investigated by future works.

**Claim 3.** *If $T_k = T_{k+1}$ we are almost surely (wrt the Lebesgue measure in the tangent space) in a local minimum of the functional $T \mapsto \sum_i d_E(T_i, T)^p$.*

**Claim 4.** *The algorithm converges in a finite number of steps.*

**Claim 5.** *The algorithm converges faster if $T_k$-maximal parallel decompositions of mappings $M_i$ are employed at every iteration.*

**Claim 6.** *Upon extending the correspondence $\mathbb{R}^{E_{T_2}} \approx \mathbf{Tan}_{[T]}(\mathcal{T})^{\|}$, $T_k$-maximal decompositions of mappings $M_i$ can be employed at every iteration, further enhancing the convergence speed of the algorithm.*

Note that we are moving $T_k$ only parallel to $\mathcal{T}^N$, that is, $dim(T_k) \leqslant dim(T_{k-1})$. Trying the algorithm for $T_0 \in \mathcal{T}^N$ for big enough $N$ and different weights initializations should bear good approximations.

## 4.8 Proofs

*Proof of* Theorem 4.

1. given $\lambda \in [0,1]$ and $T \in \mathcal{T}$ we define $\lambda \cdot T$ to be the tree obtained by shrinking each edge of $T$ by a factor of $\lambda$ i.e. if $w'$ is the weight function of $\lambda \cdot T$, we have $w'(v) = \lambda \cdot w(v)$.

   Now consider $M \in Mapp(T, T')$. We call $M_\lambda$ the same mapping but inside $Mapp(\lambda \cdot T, \lambda \cdot T')$. Since we can take $\lambda$ outside the cost of all the edits, then $cost(M_\lambda) = \lambda \cdot cost(M)$. In other words $d_E(\lambda \cdot T, \lambda \cdot T') \leqslant \lambda \cdot d_E(T, T') < d_E(T, T')$.

   This of course implies that $F : [0,1] \times \mathcal{T} \to \mathcal{T}$, such that $F(t, T) = t \cdot T$ is continuous, and so $\mathcal{T}$ is contractible.

2. given any tree $T$ and any $\varepsilon > 0$ we build the following Cauchy sequence: let $T_0 = T$ and we obtain the tree $T_n$ by attaching to a leaf of $T_{n-1}$ a pair of edges, each with length $\frac{\varepsilon}{2 \cdot 2^n}$.

   Suppose that exists $T'$ such that $T_n \to T'$.

   We know $\#V_{T_n} \to \infty$.

   Let $M_n \in Mapp(T_n, T')$ minimizing mapping. We know $cost(M_n) \to 0$. We call $C_n$ the elements of $V_{T_n}$ paired with elements of $V_{T'}$ by $M_n$.

   We know $\#C_n \leqslant K = dim(T')$. We know that all the vertices in $V_{T_n} - V_T$ have one sibling since they are added as couples. This means that, even if we take away $K$ vertices, at least one out of every pair of the remaining siblings will be deleted once $dim(T_n) > dim(T')$. We indicate the deleted vertices of $V_{T_n}$ with $D_n$.

   Of course $cost(M_n) \geqslant cost(C_n) + cost(D_n)$ and so $cost(D_n) \to 0$.

   But $cost(D_n) > \varepsilon \cdot \frac{1}{2} \sum_K^n \frac{1}{2^n}$ and so it does not go to zero.

   This shows that $C = \overline{B_\varepsilon(T)}$ is not compact. In fact $\{T_n\} \subset C$ is a Cauchy sequence but it is not converging.

3. We just need to prove local compactness and we do so via sequential compactness.

   Consider $T \in \mathcal{T}^N$ and $\{T_n\} \subset \overline{B_\varepsilon(T)}$ with $\varepsilon < min_{v \in V_T} cost(v_d)$.

   If we consider an edge $l = (v, v')$ in $T$, along the sequence $T_n$ we know that $l$ will never get wholly deleted. It might be split, shrunk but it will never disappear.

So we fix a sequence of geodesic mappings $M_n$ such that $M_n \in Mapp(T, T_n)$. For any edge $l = (v, v')$ in $T$, with $v < v'$, we consider the set $E_n^l = \{e_n^k\}$ such that $e_n^k \in M_n$ edits the edge $l$. With that we mean: shrinkings of $l$, splittings inserting points $w$ with $v < w < v'$ (note that these appear only after the shrinkings), insertion of edges in points $w$ with $v \leqslant w \leqslant v'$. Of course for each $n$, we might have $e_n^1$, $e_n^2$,... acting on the same edge. While there can be at most one shrinking on $l$ for each $n$, there can be multiple insertions or splittings; of course this number is uniformly bounded because of the dimension constraints.

For a fixed $n$ we have a natural order between splittings, given by the height at which the new point is inserted, and a similarly induced partial ordering between insertions. Insertion are comparable with respect to this partial order if they happen at different heights. We fix a random order on insertions happening at the same vertex so that all insertions are completely ordered. For a fixed $l$ we partition these edits into $Sh^n$, $S_k^n$ and $I_{k',k}^n$, which, for each $n$, collects the elements in $E_n^l$ which are respectively the shrinking, the $k$-th splitting and the $k$-th insertion at the $k'$-th inserted vertex. One can set $k' = 0$ being the index of $v$, and $k' = -1$ being the index of $v'$.

We know that for each edge $l$:

- $Sh = \cup_n Sh^n$ is at most countable, and the sequence given by the different weights of $l$ obtained through the shrinkings, is a sequence in $[L - \varepsilon, L + \varepsilon]$, with $L$ the original length of $l$. Then we can extract a converging subsequence.
- $S_k = \cup_n S_k^n$ is at most countable, then the ratio of the distance in height between $v$ and the splitting point, over the length of $l$ (after the shrinking of the $n$-th mapping) form a sequence in $[0, 1]$. So we can extract a converging subsequence in $[0, 1]$. In other words we can find a subsequence of edits which converges to a certain splitting.
- $I_{k',k} = \cup_n I_{k',k}^n$ is at most countable, then the insertions at $v$, form a sequence in $[0, L]$, and the length of the inserted edges form a sequence in $[0, \varepsilon]$. So we can extract a converging subsequence.

If some of the sets defined above are not countable we discard from the sequence the indexes $n$ which appear in the collection of edit, being it $Sh$, $S_k$ or $I_{k,'k}$. Note that, by the dimension bounds, we have a finite number of such sets, for a finite number of edges.

Thus, for every edge $l$, we proceed in this way:

(a) since $Sh$ is countable we extract a converging subsequence of shrinkings.

(b) starting from $k = 1$, from the subsequence just obtained we extract a converging subsequence of splitting locations. Starting from this sequence we

repeat the extraction procedure for $k = 2$, then recursively till we reach $k$ such that $S_k$ is empty. Again if, for some $k$, $S_k$ is finite, we simply discard elements of the subsequence appearing in $S_k$.

(c) lastly, from the last set if indexes we obtained, we extract from $I_{k',k}$ a converging subsequence of insertions for every $k$ and $k'$; working on $k$ and $k'$ as in the previous point. In a finite number of steps we reach $I_{\hat{k'},\hat{k}}$ such that for all $k' > \hat{k'}$ and $k > \hat{k}$, $I_{k,k'} = \varnothing$.

Given any ordering on the edges, we recursively apply this for every $l$.

Thus, taking for every edge and for a fixed $n$ the edits in $Sh^n \cup_k S_k^n \cup_{k',k} I_{k',k}^n$ along the final subsequence of merge trees, we obtain a sequence of mappings $N_n$, defined on $T$, each contained in $M_n$. Let call $ST_n$ the tree obtained from $T$ with the edits in $N_n$. Then clearly $N_n \in Mapp(T, ST_n)$. Moreover, $T_n$ can be obtained from $ST_n$ with the edits in $M_n - N_n$, which are only insertions.

By construction, any edit in $N_n$ is part of a converging sequence of edits. We call $\overline{N}$ the mapping obtained with the limit of the edits of $N_n$. Let $ST$ be the tree obtained from $T$ with $\overline{N}$. We have $ST_n \to ST$. In fact, consider one edge $l \in E_T$. Take for instance the sequence of shrinkings in $Sh = \{e_1, \ldots, e_n\}$ parametrized such that $w_{ST_n}(l) = e_n$. For any fixed $\epsilon$ and for $n$ big enough we know $|e_n - e_{n+1}| < \epsilon$. Thus $|w_{ST_n}(l) - w_{ST_{n+1}}(l)| < \epsilon$ is a shrinking with cost less then $\epsilon$. For the same edge $l$ there are at most $N$ splittings, each edit splits $l$ in $E_{ST_n}$ at a certain height. The difference in heights between splittings in $S_k^n$ and $S_k^{k+1}$ is going to zero, and thus we can again choose $n$ big enough so that the $k-$th splitting of $ST_n$ can be turned in the $k-$th splitting of $ST_{n+1}$, for all $k$, with cost less than $\epsilon$.

The same reasoning can be done on the weights of the insertions. Thus we can go from $ST_n$ to $ST_{n+1}$ with a finite set of edits, with cost less that $\epsilon$ and whose number is uniformly bound. For instance we know that on every tree we can have at most $N$ shrinkings, $N$ ghostings, and $N$ insertions. Thus,for every $\epsilon$, there is $n$ big enough such that $d_E(ST_n, ST_{n+1}) < 3N\epsilon$.

Working always with the subsequence obtained in the previous steps, we take each $M_n$ and we substitute each edit in $N_n \subset M_n$ with its limit $\overline{N}$.

We can obtain a new sequence of trees $T'_n$ and mappings $M'_n \in Mapp(ST, T'_n)$ in this way:

- $T'_n$ is obtained from $T_n$ taking the limit on $ST_n \to ST$.
- $M'_n$ is the mapping given by the identity on the subtree $ST \subset T'_n$, and the edits in $M_n - N_n$, which, as already noted, are only insertions.

We know that, since $cost(M_n - N_n) \leqslant cost(M_n) < \varepsilon$, $\{T'_n\} \subset \overline{B_\varepsilon(ST)}$; of course we do not know if the relationship $\varepsilon < min_{v \in V_T} cost(v_d)$ holds also for $ST$, but

nevertheless we know by construction that on $ST$ there are no more shrinking and deletions to be done by any $M'_n$.

So we can do the same steps performed up to now on this other sequence, obtaining a subsequence of $\{T'_n\}$ with subtrees $ST'_n \subset T'_n$ such that $ST'_n \to ST'$ and with mappings $N'_n \in Mapp(ST'_n, ST')$, $N'_n \subset M'_n$. If $N'_n = M'_n$ then $ST'_n = T'_n$ and viceversa.

By construction $id_{ST} \subset M'_n$, where with $id_{ST}$ we mean a mapping which doesn't delete, ghost or change weight to any vertex of $ST$; $id_{ST}$ is equal to $N'_n$ iff $Sh$, $S_k$ and $I_{k',k}$ are empty for all the edges of $T'_n$ , and so $T'_n = ST$; otherwise $id_{ST} \subsetneq N'_n$. This means that there exists a limit for the edits in $N_n \cup (N'_n - id_{ST}) \subset M_n$ and so a limit for a sequence of trees $SST_n$, $T_n$ obtained from $SST_n$ with insertions, with $E_{SST_n}$ which is either strictly bigger than $E_{ST_n}$ or equal to $E_{ST_n} = E_{T_n}$.

Being the number of edges in $T_n$ uniformly bound, in a finite number of steps we can extract a converging subsequence of $\{T_n\}$.

■

*Proof of* Theorem 5.

Consider a Cauchy sequence $\{T_n\}_{n\in\mathbb{N}}$. By Proposition 9, $\big|||T_n|| - ||T_m|||\big| \leqslant d_E(T_n, T_m)$. Thus, $\{||T_n||\}$ is a Cauchy sequence and thus, it converges in $\mathbb{R}$. In other words, $||T_n|| \to C$. If $C = 0$ then $T_n \to 0$, the tree with one vertex and no edges. Therefore, we can suppose $C > 0$.

Define $\epsilon(T) := \min_{e\in E_T} cost(e_d)$. In the proof of Theorem 4 we show that $\overline{B_r(T_m)}$ is compact for every $r < \epsilon(T_m)$. We know $\{\epsilon(T_n)\}$ is a bounded sequence in $\mathbb{R}$ and thus, up to taking a subsequence, it converges. If $\epsilon(T_n) \to q > 0$ then, the sequence $\{T_n\}$ is definitely in a compact ball centered in some $\widehat{T}_m$, obtained from $T_m$ by possibly raising the weight of its maximal edges, so that $\epsilon(T_m) < \epsilon(\widehat{T}_m)$ and $d_E(T_m, \widehat{T}_m) < \epsilon(\widehat{T}_m)$. Thus $\{T_n\}$ converges.

Suppose then $\epsilon(T_n) \to 0$. For all $n$, take $T_n$, and obtain $T^1_n$ by deleting $\operatorname{argmin}_{e\in E_{T_n}} cost(e_d)$. By construction we know $d(T_n, T^1_n) \to 0$. Thus if $\{T^1_n\}$ converges, also $\{T_n\}$ converges. We repeat the same reasoning as above, considering $\{\epsilon(T_n)\}$; if $\epsilon(T^1_n) \to q > 0$ we are done, otherwise we take $\{T^1_n\}$ and obtain $\{T^2_n\}$ removing the smallest edge and repeat again the procedure. Since $\#E_{T_n} \leqslant N$, we know $\#E_{T^1_n} \leqslant N-1$ and so $\#E_{T^j_n} \leqslant N-j$ etc. Since $T^j_n \to C$, $0 < \#E_{T^j_n}$ and thus in a finite number of step we obtain $\{T^j_n\}$ which converges and so does $\{T_n\}$.

■

*Proof of* Proposition 10.

Completeness is easily obtained because, given a Cauchy sequence $\{T_n\}$ with $dim(T_n) < N$ and $||T_n|| \leqslant C$, then by Theorem 5 and Proposition 9 we have: $T_n \to T$ and $||T_n|| \to ||T|| \leqslant C$.

Now we prove the second statement of the proposition. For a fixed $N$ we only have a finite number of available tree structures (without order two vertices). The weight of every edge in such tree structures is bound to be in $(0, C]$. Fix $\varepsilon > 0$ and take a sequence of numbers $0 = a_1 < \ldots < a_M = C$ such that $a_{i+1} - a_i < \varepsilon/N$. Let Consider $w_i = (a_{i+1} + a_i)/2$.

For each of the available tree structures, take all possible combinations of weights $w_1, \ldots, w_N$ in the edges. For instance, if we consider the tree structure $T$ with just two vertices $a < b = r_T$ and one edge $e = (a, b)$; then we have $N$ possible merge trees given by $w_T(e) = w_i$. When we have three vertices and two edges $E_T = \{e, e'\}$ we need to take all the combinations $w_T(e) = w_i$ and $w_T(e') = w_j$ and so on and so forth. We call this finite set of merge trees $\mathcal{A}_\varepsilon$.

Take now a merge tree $T$ with $dim(T) \leqslant N$ and $||T|| < C$. There is at least one tree $T' \in \mathcal{A}_\varepsilon$ and such that there is $g : V_T \to V_{T'}$ isomorphism of tree structures with $|w_T(e) - w_{T'}(g(e))| < \varepsilon/N$. Which implies $d_E(T, T') < \varepsilon$.

■

*Proof of* Proposition 11.

It is enough to attach to any of the leaves of $T$ a pair of equal branches of length less than $\varepsilon/2$ each. Deleting both edges or deleting one, ghosting the vertex and shrinking the other are both geodesics.

■

*Proof of* Lemma 4.

Consider $M$ minimizing mapping for $d_E(\overbrace{V_{i_1}, ..., V_{i_k}}, \overbrace{V_{j_1}, ..., V_{j_h}})$, we will call $r$ and $r'$ their roots.

Consider $\Gamma_M^{(r,r')}$. If $M$ does not contain deletions, for any couple $(V, \Gamma_M^{(r,r')}(V))$ we have $V = sub_i(r)$ and $\Gamma_M^{(r,r')}(V) = sub_j(r')$, so clearly $cost(M) > \varepsilon$.

Otherwise we have at least one deletion, with cost at least $\varepsilon$.

■

*Proof of* Corollary 5.

1. Let $Sub(r_T) = \{V_1, ..., V_n\}$. Suppose:

$$\Gamma_M^{(r_T, r_{T'})}(W)^{-1} = \overbrace{V_{i_1}, ..., V_{i_k}}$$

$$\Gamma_{M'}^{(r_T, r_{T'})}(W)^{-1} = \overbrace{V_{j_1}, ..., V_{j_h}}$$

   We know $\overbrace{V_{i_1}, ..., V_{i_k}} \neq \overbrace{V_{j_1}, ..., V_{j_h}}$ and so by the definition of $K_T$ we can apply Lemma 4.

2.
$$cost(M) + cost(M') < 2K_T$$

$$cost(M)+cost(M') = \sum_{W \in Sub(r_{T'})} d_E(\Gamma_M^{(r_T,r_{T'})}(W)^{-1}, W)+d_E(\Gamma_{M'}^{(r_T,r_{T'})}(W)^{-1}, W)$$

Suppose exists $W \in Sub(r_{T'})$ such that $\Gamma_M^{(r_T,r_{T'})}(W)^{-1} \neq \Gamma_{M'}^{(r_T,r_{T'})}(W)^{-1}$. Then:

$$d_E(\Gamma_M^{(r_T,r_{T'})}(W)^{-1}, W) + d_E(\Gamma_{M'}^{(r_T,r_{T'})}(W)^{-1}, W) > 2K_T$$

which is absurd.

■

*Proof of* Lemma 5.

1. First we prove that there can be at most one edge (or sequence of consecutive edges) of $T'$ that goes from coupled to a single edge of $T$ in $M$, to deleted in $M'$, or that goes from being deleted in $M'$, to being coupled to a single edge of $T$ in $M$.

   Consider two edges (or sequences of adjacent edges) of $T'$ of length $a$, $b$ which in $M$ are coupled with edges of $T$ with length $A$ and $B$ respectively contributing to the distance with $|A-a|+|B-b|$. If in $M'$ they are deleted, the contribute to that cost bt $a+b$. This situation gives the following set of equations:

   $$a, b, A, B > 0$$

   $$A, B > K_T > 0$$

   $$|A-a|+|B-b| < K_T$$

   $$a+b < K_T$$

   This system of course has solution only if $K_T > a$ and $K_T > b$, and so it becomes:

   $$a, b, A, B > 0$$

   $$A, B > K_T > 0$$

   $$A+B-K_T < a+b$$

   $$a+b < K_T$$

   which is impossible since it gives:

   $$2K_T < A+B < 2K_T$$

   The roles of $M$ and $M'$ can of course be reversed.

2. Now we prove the thesis of the Lemma.

   Consider $M, M' \in Mapp(T', T)$ minimizing mappings.

   Suppose there is an edge $[w, w']$ (or let $w < w'$ be extremes of a sequence of adjacent edges, which become an edge after deletions and ghostings) in $T'$ such that $(w, D) \in (M' - M)$, and such that $(w, x) \in M$ for some $x \in V_T$. By the previous point we know that this is not happening for any other edge or sequences of adjacent edges.

   We note that, by hypothesis, $w'$ and $w$ are not ghosted by $M$. To lighten the notation, here we call $M_D$ the set of deletions in a mapping $M$.

   Apply on $T'$ all the deletions in $M_D \cap M'_D$ to be applied on it, obtaining $T''$. We induce in a natural way mappings $N$ and $N'$ from $T''$ to $T$, simply removing from $M$ and $M'$ the deletions already done.

   All the edges still to be deleted by $N$, cannot be paired with any other edge by $N'$ (nor can they be deleted, since these deletions do not lie in $M_D \cap M'_D$). So such edges are left by $N'_D$ with one of the two extremes of order 2.

   To recap, we start from $T''$ and each deletion we have in $N_D$ or $N'_D$ deletes an edge which is left with at least one order 2 vertex by the other mapping. For both mappings there are no insertions to be done to obtain $T$, because their cost would be over $K_T$.

   Consider $(v, v')$ edge in $T''$ which is deleted by $N$ and not by $N'$. Suppose $v'$ is of order two after $N'_D$. Let $v_1, .., v_n$ be the children of $v'$ in $T''$. We know that $sub_i(v')$ is deleted by $N'_D$ for every $i$ but one, be it $h$, such that $v_h = v$. This in turns tells us that $sub_i(v')$ with $i \neq h$ are not deleted by $N$ and all their edges must have at least one vertex of order two. Having removed all ghost vertices in $sub_i(v')$, all the edges remaining pass from being deleted to being coupled. This means that $i \leqslant 2$, $w' = v'$, $v_1 = v$ and $v_2 = w$.

   Notice that, by supposing that $v$ instead of $v'$ is of order 2 after $N'_D$, we can repeat the same reasoning for some edge $(v'', v)$ for which $v$ is the extreme closer to the root.

   By the uniqueness of $[w, w']$, we know that for any other edge of $T''$ deleted by $N$ and not by $N'$, the extreme of order two after $N'_D$ can have no siblings in $T''$, i.e. they are already of order two. And the same reversing the role of $N$ and $N'$.

   So, apart from the deletion of $(w, w')$ by $N'_D$, the others in $N_D$ or $N'_D - \{w_d\}$ provide no changes in the tree structures, up to order $2$ vertices.

   So the tree structure obtained from their deletions or the one resulting from the ghosting of their order two extremes is the same, up to order two vertices.

   Since $T$ has no order 2 vertices, and since there are no insertions to be done on $T''$, $[w, w']$ is paired with an edge with no vertices of order two, and so its deletion change the topology of the tree $T''$.

In other words we obtain the same tree structure, which is the one of $T$, both by removing from $T''$ the order 2 vertices, and from first removing the order 2 vertices and then deleting the internal edge $[w, w']$ which is absurd.

■

*Proof of* Theorem 6.

Suppose we have $M$ and $M'$ minimizing mappings. We know $\Gamma_M^{(r_T, r_{T'})} = \Gamma_{M'}^{(r_T, r_{T'})}$ by Corollary 5; so to lighten the notation we will just write $\Gamma$.

First suppose $\Gamma(sub_i(r_T)) = sub_j(r_{T'})$. We shall call $V = sub_i(r_T)$ and $W = sub_j(r_{T'})$ .

Let $v$ be the only child of $r_T$ belonging to $V$. Since we have no deletions on $T$, $v$ must be assigned to a vertex of $T'$.

Let the couple $(v, w) \in M$ and the couple $(v, w') \in M'$, so that the edge $[v, r_T]$ is shrunk to $[w, r_{T'}]$ by $M$, and to $[w', r_{T'}]$ by $M'$. All after deletions and ghostings.

If $v$ is a leaf, we have nothing to prove. Thus, suppose $v$ is not a leaf. The first claim we prove is that either $w > w'$, or $w' > w$ or $w = w'$ in $T'$, according to the partial order relationship given by $father > son$.

Suppose this does not happen. This means that $w$ and $w'$ are not on the same path from some leaf to the root. In other words $sub(w)$ and $sub(w')$ have empty intersection. Since $sub(v)$ is mapped in $sub(w)$ and $sub(w')$ respectively by $M$ and $M'$, we know that $sub(w')$ and $sub(w)$ must be deleted by $M$ and $M'$ respectively. However, $sub(w')$ and $sub(w)$ are assigned in the other mapping. This contradicts Lemma 5.

Now we prove $w = w'$. If $w \neq w'$, we know $w > w'$ or $'w > w$ holds. Suppose $w >' w$. Thus, this means that $w$ is on the path from $w'$ to $r_T$. This implies that in $M'$, $w$ is either ghosted or deleted.

Suppose it is ghosted: then $M'$ must delete all but one children of $w$. Since $v$ is of order greater than two, and $M$ assigns $v$ to $w$, there are edges which go from assigned in $M$ to deleted in $M'$, contradicting Lemma 5. However, for the same reason $w$ cannot be deleted by $M'$ coupled by $M$.

Thus, we conclude that $w = w'$.

Since $w = w'$, we are in the position to apply all this machinery on $sub(v)$ and $sub(w)$ (until we reach the leaves) if the hypothesis $\Gamma(sub_i(r_{sub(v)})) = sub_j(r_{sub(w)})$ holds again. In that case, we end up with $M$ equal to $M'$ on all internal vertices.

Now we need to reduce the general case to the case $\Gamma(sub_i(r_T)) = sub_j(r_{T'})$.

Suppose $\Gamma(\overbrace{sub_{i_1}(r_T)...sub_{i_h}(r_T)}) = sub_j(r_{T'})$.

We shall call $V = \overbrace{sub_{i_1}(r_T)...sub_{i_h}(r_T)}$, $V_{i_k} = sub_{i_k}(r_T)$, and $W = sub_j(r_{T'})$.

This hypothesis means that, both for $M$ and $M'$, after the deletions $r_W$ (the root of $W$) has $h$ children. Consider $M_D \cap M'_D$. We call $W'$ the tree obtained from $W$ and applying the deletions of its internal edges appearing in $M_D \cap M'_D$. Then we consider $N$ and $N'$, the mapping induced by $M$ and $M'$ between $V$ and $W'$. They are well defined since we obtained $W'$ applying deletions contained in both mappings. By Lemma 5 we

know we are left only with deletions of order two vertices, the order of the vertices will not change in any of the mappings. That is, up to order two vertices, the tree structures of $W'$ is isomorphic to the one of $V$.

This in particular means that the root of $W'$ has order $h$ since $r_V$ has order $h$, and that $\Gamma(sub_i(r_V)) = sub_j(r_{W'})$. Therefore, by the first part of the proof, $N = N'$ on internal edges, and consequently, on such vertices, $M = M'$. ■

*Proof of* Proposition 12.

Since $\mathcal{F}(T) = \sum_i d_E^2(T, T_i)$ is a continuous real valued function, if we can restrict the minimization domain in some compact subset of $\mathcal{T}$, we obtain the result, since continuous functions preserve compactness.

First we know that, if $\overline{T}$ exists, then $||\overline{T}|| \leqslant \sum_i ||T_i||^2$. Otherwise $\mathcal{F}(0) < \mathcal{F}(\overline{T})$, with $0$ being the tree with no edges.

Lets call $N_i = dim(T_i)$ and $N = \sum_i N_i$. Consider $T$ such that $\#L_T = R$ with $R > N$. Then, for all $i$, any geodesic between $T$ and $T_i$ deletes at least $R - \#L_{T_i}$ edges of $T$. Since $\#L_{T_i} \leqslant N_i$, we have $\sum_i R - \#L_{T_i} > \sum_i R - \#N_i > R(n-1)$. This implies that there is at least one leaf of $T$ which is deleted all the times. However, then, if we delete it, we obtain $T'$ such that, for all $i$, $d_E(T', T_i) < d_E(T, T_i)$.

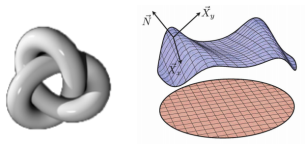
Thus, the number of leaves of $\overline{T}$ cannot exceed $N$. But this immediately implies that, if it exists, $dim(\overline{T}) < 2N$.

Since we have a bound on the norm and the dimension of $\overline{T}$, we can restrict the optimization domain on a compact set, which means that $\overline{T}$ exists.

■

CHAPTER 5

# Further Directions for Tree-Like topological summaries



In this chapter we present some possibile future developments of the content of Chapter 2, Chapter 3 and Chapter 4. These research directions are very diversified and include both possible generalizations of the topological summaries introduced in Chapter 2 and ways to exploit the objects defined in Chapter 4 to further develop tools to work with sets of generalized dendrograms.

## 5.1 Further Comparisons with other Metrics for Trees and Merge Trees

The edit distance presented and employed in Chapter 2 and Chapter 3 is novel and designed with the aim of comparing topological information whilst retaining a feasible complexity. Throughout the previous chapters some very important differences with already existing metrics defined for merge trees have been stressed. However, we

think that a deeper comparison with other edit distances for Reeb Graphs, interleaving distances, Gromov-Hausdorff related distances between ultrametrics and possibily other more classic metrics for trees could help further highlighting the strengths and weaknesses of our approach and establish relationships between the different theoretical frameworks.

## 5.2 Stability issues

As highlighted in Chapter 3, the metric $d_E$ is quite sensitive to large amount of noise and, in general, to the size of the tree: large trees (in terms of number of leaves), tend to have bigger distances, because they need more edits to be modified. Going in the direction of Section 5.1, we see that interleaving distances (De Silva et al., 2016; Gasparovic et al., 2019; Morozov et al., 2013) solve this problem by "moving simultaneously" the whole tree by the same amount along the real line, obtaining a more "stable" metric: the operator which maps functions into summaries is 1-Lipschitz. As pointed out by Remark 4, such approach is not naively applicable with our edit distance, because splitting edges allows for simultaneous movement of more pieces of the tree, shortening paths at will. However, the pruning operator $P_\varepsilon$ suggests that the following definition could be very suited for handling noise (that is, a possibly big number of leaves with small weights):

$$d_P(T, T') := \int_{[0,+\infty)} d_E(P_\varepsilon(T), P_\varepsilon(T'))d\mu(\varepsilon)$$

where $\mu$ is a finite measure on $\mathbb{R}$. Since pruning a tree simultaneously removes noisy branches with weight below some threshold, we are going towards the idea of interleaving distance; the noise disappear after a low value of $\varepsilon$, and its contribution to the final distance can easily be controlled with $\mu$.

Note that if we work with functions with bounded total persistence (with the degree $k$ total persistence of a diagram $D$ being the sum $\sum_{p\in D} pers^k(p)$, with $p$ being the points of the diagram, and $pers(p)$ being their persistence - see Cohen-Steiner et al. (2010)), with, for instance, degree 1 total persistence being less or equal some constant $C > 0$, then for any fixed $\varepsilon > 0$, we know that any persistence diagram obtained from such functions can have at most $C/\varepsilon$ features with persistence equal or greater than $\varepsilon$. Since every feature in the persistence diagram is associated to a leaf of the corresponding merge tree $T$ and the persistence of a feature $v$ is always equal or lower than the weight $w_T(v)$, if we compose the merge tree operator $f \mapsto T_f$ with a fixed pruning operator $P_\varepsilon$, we obtain a continuous operator wrt the sup norm for functions and the edit distance for merge trees.

Another possible approach which could be pursued to obtain a more "stable" metric is to try mimicking the interleaving distance in a different way and at the same time avoiding the situation showcased in Remark 4. In this sense we think that a sensible

definition to be considered could be the following one. We recall that $L_T$, with $T$ being a tree structure, is the set of leaves of $T$ and $\zeta_v$, with $v$ being a vertex of $T$, is the ordered set $\{v' \in V_T | v' \geqslant v\}$. Moreover, for each vertex $v$ in $V_T$, we know that a mapping induces a unique edit for $v$; thus we can indicate with $cost(v)$ the cost of the edit associated to $v$ by the mapping.

**Definition 32.** *Consider $T$, $T'$ merge trees and let $M \in Mapp(T, T')$. Then:*

$$||M||_\infty = max_{l \in L_T \bigcup L_{T'}} \sum_{v \in \zeta_l} cost(v)$$

**Claim 7.** *The rule $d_\infty(T, T') := min_{M \in Mapp(T,T')} ||M||_\infty$ defines a metric for merge trees identified up to order 2 vertices.*

Instead of adding all the local contributions of the cost of turning $T$ into $T'$, with $d_\infty$ we are in some sense capturing the least amount of editing we need to do on each "branch" $\zeta_l$ (with $l$ leaf), to turn $T$ into $T'$. Thus, instead of considering the whole tree, we are limiting ourselves to "branches" considered singularly.

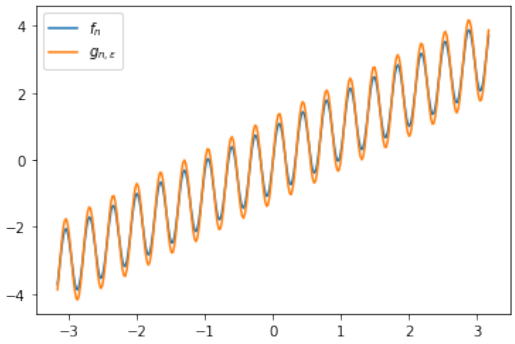
We think that the following example motivates this research path. Suppose we have $||f_n - g||_\infty < \varepsilon$; with $\#V_{T_{f_n}} = n$ and, for simplicity, suppose $T_{f_n}$ has a full binary tree structure. We know that, for a full binary tree structure, if $n$ is the number of leaves, the number of edges of the tree grows like $n$, while the cardinality of $\zeta_v$, for some leaf $v$, grows like $log_2(n)$. Thus, for $n > \#V_{T_g}$, from Theorem 3 we know that $d_\infty(T_{f_n}, T_g) \leqslant 2\varepsilon log_2(n)$, which is a number that grows much slower than the bound for the edit distance, and can be easily bound using bounded total persistence.

Exploiting the following claim, we are also able to code a promising hands-on example.

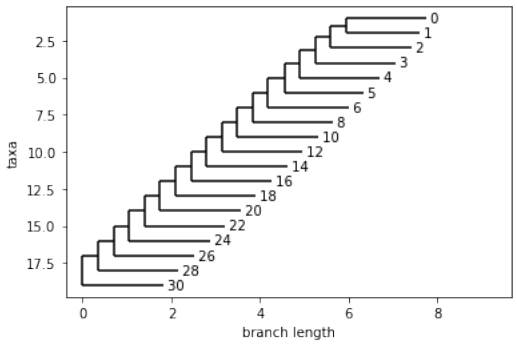
**Claim 8.** *The metric $d_\infty$ sastisfies the same decomposition properties as $d_E$.*

Consider the functions $f_n(x) = sin(2nx) + x$ defined on $[-\pi, \pi]$ and the functions $g_{n,\varepsilon}(x) = (1 + \varepsilon) \cdot sin(2nx) + x$, with $n \in \mathbb{N}$ and $\varepsilon > 0$. See Figure 5.2(a). Note that $||f_n - g_{n,\varepsilon}||_\infty = \sup_{[-\pi,\pi]} \varepsilon sin(2nx) = \varepsilon$. In Figure 5.2(b) we can see an instance of $T_{f_n}$. We compute $d_\infty(T_{f_n}, T_{g_{n,\varepsilon}})$ and $d_E(T_{f_n}, T_{g_{n,\varepsilon}})$ to compare their evolution as $n$ grows. Note that the total persistence of $f_n$ and $g_{n,\varepsilon}$ is unbound as $n$ grows and, since $\#V_{T_{f_n}} = \#V_{T_{g_{n,\varepsilon}}} = 4n + 1$, using Theorem 3 we have $d_E(T_{f_n}, T_{g_{n,\varepsilon}}) \leqslant 2(4n + 1)^2\varepsilon$. Moreover, since $\#\zeta_{v_0} = \#\zeta_{w_0} = 2n$ (with $v_0$ and $w_0$ being the leaves associated to the minimum of $f_n$ and $g_{n,\varepsilon}$ respectively), we have, again through Theorem 3, $d_\infty(T_{f_n}, T_{g_{n,\varepsilon}}) \leqslant 4n\varepsilon$. We see from Figure 5.2(c) that these bounds are very large and indeed $d_E$ shows a linear growth and not a quadratic one, while $d_\infty$ quickly flattens and is below even a logarithmic growth.

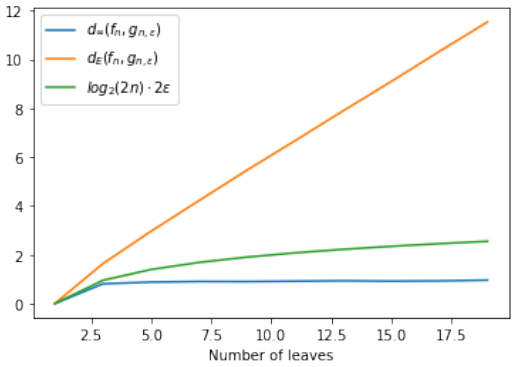
## 5.3 Tangent Structure and Statistical Tools




**(a)** *Plot of $f_n$ and $g_{n,\varepsilon}$ for $n = 8$ and $\varepsilon = 0.3$.*




**(b)** *The merge tree $T_{f_n}$, for $n = 8$.*




**(c)** *Growth of the different metrics.*

**Figure 5.2:** *Plots referring to the hands-on example showcasing the potential of the definition of $d_\infty$.*

As evident from Chapter 4, the dendrogram spaces are in general not very well behaved in terms of geometric properties. Thus there are not many naive statistical tools that can be employed. However, at least for merge trees, there some particular properties, like the decomposition of mappings and the linear structure *parallel* to the strata $\mathcal{T}^N$ that can be employed to capture some features of finite sets of merge trees. For instance a "zero dimensional" summary given by a Frechét mean. A natural step forward in this direction would be the definition of higher dimensional summaries, according to some notion of dimension, which would allow the definition of a PCA technique in such space.

In a similar fashion, the local linear *parallel* structure could be exploited to build some linear models in the tangent space. A major obstacle to be faced, in this sense, is the definition of a *log* map which can move merge trees into a tangent space. The definition of such map is very delicate due to the presence of multiple minimizing geodesics, which imply that the exponential map is non injective for large sets of merge trees. Trying to define an inverse for the exponential map very likely causes issues with the continuity of such inverse. Another issue which must be overcome is that the standard inner product is naturally bound with the 2-norm $||\cdot||_2$ in $\mathbb{R}^N$, which is not compatible with the edit distance $d_E$.

Along with that, we are not completely satisfied by the tangent structure defined in Chapter 4, which has some theoretical drawbacks and is quite convoluted. We think that some tweaks in the definitions and notation in Section 4.6 could lead to clearer discussions, proofs and results.

## 5.4 Locally & Weakly Editable Spaces and Multipersistence

One of the most natural directions in which the content presented in Chapter 2 can be expanded is the one of generalizing the set of weights for which the metric $d_E$ is well defined and computable. We propose two definitions which include some interesting kind of spaces and which are likely to still induce an edit distance with good properties.

**Definition 33.** *A set $X$ is called weakly editable if the following conditions are satisfied:*

*(P1)* $(X, d)$ *is a metric space*

*(P2)* $(X, \oplus, 0)$ *is a monoid (that is $X$ has an associative operation $\oplus$ with zero element 0)*

*(P3)* $d(x \oplus y, 0) \leqslant d(0, x) + d(0, y)$*, that is splitting edges lenghten deletions*

*(P4)* $d(x, y) + d(v, w) \geqslant d(v \oplus x, w \oplus y)$ *and* $d(x, y) + d(v, w) \geqslant d(x \oplus v, y \oplus w)$*, that is, ghosting vertices shortens shrinkings.*

Weakly editable spaces include: normed spaces, the set of finite Sets, with $d(A, B) = \#(A \cup B) - \#(A \cap B)$ and the space of persistence diagrams with the operation given by the disjoint union of sets of points.

These families of spaces account for some quite interesting approaches. We can put labels on leaves of dendrograms, which for instance can be used to compare clustering structures on the same set of labels. The set of labels could be only partially overlapping. Similarly this could open up the possibility of working with left-right ordered tree structures, which make sense in particular for merge trees representing functions defined on the real line.

When working with merge trees of functions, we could also record the point giving birth to a connected component (that is, the local minimum) as well as many other kind of data. This situation allows to play with the invariance properties of merge trees of functions, since the more information we embedd on the tree-like representation of the function, the more the class of functions represented by the same topological summary, shrinks. Lastly, using persistence diagrams to induce weights on edges can introduce a novel approach to multipersistence: we could use a function to build the dendrogram and (use another function to) extract persistence diagrams on the connected components induced by the first function.

We consider also this second definition.

**Definition 34.** *Given a $X$ weakly editable and a set $\mathcal{S}$ of dendrograms with $X$-valued multiplicity functions, $X$ is called $\mathcal{S}$-locally editable if $\varphi(T)$ is an editable subset of $X$, for every $T \in \mathcal{S}$.*

Examples of locally editable spaces can easily be obtained by taking curves into weakly editable spaces as in Section 2.5. We have seen that such framework very well suited to record valuable information about a fixed basis along along a fixed basis vector spaces filtration.

**Claim 9.** *If $X$ is a weakly editable space, then $d_E$ is a metric for $(\mathcal{D}_2, X \coprod Y)$. Moreover, if $X$ is locally editable on $\{T, T'\}$, then $d_E(T, T')$ can be computed with the algorithm presented in Section 2.7.*

## 5.5 Reeb Graphs

Recently Stefanou (2020) proposed a decomposition of Reeb Graphs in terms of ordered sets of merge trees. This could open up the possibility of defining a metric framework on Reeb Graphs starting from merge trees. Many aspects of this possibility should be investigated, but probably the most important ones are:

- the interpretability: the way in which distances between sets of merge trees become distances between Reeb Graphs must be reasonable;
- can the decomposition property be extended to work with Reeb Graphs with more general multiplicity functions?

Even if this decomposition fails to be of any practical interest for our metric $d_E$, we believe that upon adding enough variables to take into account the possible cycles contained into Reeb Graphs, our edit distance, along with the proposed algorithm, could be extended to work also with such more general objects.

## 5.6 Total Variation of Functions

The Edit Distance defined for Reeb Graphs in Bauer et al. (2016); Di Fabio and Landi (2012, 2016) posseses a very elegant characterization in terms of $||\cdot||_\infty$ between continuous functions on a manifold (with some further hypotheses), up to homeomorphic reparametrization. We think that, at least in the case of curves, our metric should behave similarly with respect to the total variation of a function.

## 5.7 Stability properties in applications

As done in Chapter 3, when applying the framework discussed in Chapter 2, the continuity properties of the operator assigning dendrograms to data, must be carefully studied, because from such properties depends the interpretability of the results.

Consider for instance the following proposition.

**Proposition 28.** *Let $C = \{x_0, \dots, x_n\}$ and $C' = \{y_0, \dots, y_n\}$ two point clouds in $\mathbb{R}^n$, with $x_i = y_i$ for all $i > 0$. And let $\rho = d(x_0, y_0)$.*

*Consider the trees $T_C$ and $T_{C'}$ which are the single linkage hierarchical dendrograms obtained from the point clouds $C$ and $C'$ respectively.*

*Then $d_E(T_C, T_{C'}) \leqslant 4n\rho$.*

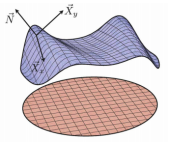
*Proof.* We obtain this result by applying Theorem 3.

Consider the simplicial complex $A$ obtained with the 0-simplexes $\{a_0, \dots, a_n\}$ and the complete set of 1-simplexes $\{[a_i, a_j]\}$. Define $f, g : A \to \mathbb{R}$ such that $f(a_i) = g(a_i) = 0$, $f(a_0, a_j) = d(x_0, x_j)$ and $g(a_0, a_j) = d(y_0, x_j)$. Then, the merge trees $T_f$ and $T_g$ coincide with $T_C$ and $T_{C'}$; moreover $||f - g||_\infty = \rho$. By Theorem 3 we conclude the proof. □

This suggest that $d_E$ has some continuity properties when applied to point clouds and hierarchical dendrograms. In turn, this points out that the metric $d_E$ could be used to compare, for instance clustering structures. It also indicates that there might be similar results in other kinds of applications.

CHAPTER 6

# Projected Methods in 1-D Wasserstein Spaces



The content of this chapter is also part of the paper Pegoraro and Beraha (2021).

In this chapter we present a novel class of *projected* methods to perform statistical analysis on a data set of probability distributions on the real line, with the 2-Wasserstein metric. We focus in particular on Principal Component Analysis (PCA) and regression. To define these models, we exploit a representation of the Wasserstein space closely related to its weak Riemannian structure, by mapping the data to a suitable linear space and using a metric projection operator to constrain the results in the Wasserstein space. By carefully choosing the tangent point, we are able to derive fast empirical methods, exploiting a constrained B-spline approximation. As a byproduct of our approach, we are also able to derive faster routines for previous work on PCA for distributions. By means of simulation studies, we compare our approaches to previously proposed methods, showing that our *projected* PCA has similar performance for a fraction of the computational cost and that the *projected* regression is extremely flexible even under misspecification. Several theoretical properties of the models are investigated, and

asymptotic consistency is proven. Two real world applications to Covid-19 mortality in the US and wind speed forecasting are discussed.

## 6.1 Introduction

In many fields of machine learning and statistics, performing inference on a set of distributions is an ubiquitous but arduous task. The Wasserstein distance provides a powerful tool to compare distributions, as it requires very little assumptions on them and is at the same time reasonably easy to compute numerically. In fact, many other distances for distributions either require the existence of a probability density function or are impossible to evaluate, cf. Cuturi (2013), Peyré et al. (2019), Panaretos and Zemel (2020).

The Wasserstein distance recently gained popularity both in the statistics and machine learning community. See for instance Bassetti et al. (2006), Bernton et al. (2019a), Catalano et al. (2021) for statistical properties of the Wasserstein distance, Cao et al. (2019), Cuturi et al. (2019) and Cuturi and Doucet (2014) for applications in the field of machine and deep learning, Bernton et al. (2019b) and Srivastava et al. (2015) for applications in Bayesian computation.

In this work, we focus on the situation in which the single observation itself can be seen as a distribution, as in the analysis of images (Banerjee et al., 2015; Cuturi and Doucet, 2014), census data (Cazelles et al., 2018), econometric surveys Potter et al. (2017) and process monitoring (Hron et al., 2014). In particular, we consider observations to be distributions on the real line. There exist several possible ways to represent distributions, such as histograms, probability density functions (pdfs) and cumulative density functions (cdfs), each characterized by different constraints. For instance, histograms sum to one, pdfs integrate to one, and the limits for cdfs are 0 and 1, moreover all of these functions are nonnegative. These constraints translate into complex geometrical structures that characterize the underlying spaces these objects live in.

### 6.1.1 Previous work on distributional data analysis

One of the first works defining PCA for a data set of distributions is Kneip and Utikal (2001), where the authors apply tools from functional data analysis (FDA) directly to a collection of probability density functions. This approach, however, completely ignores the constrained nature of probability density functions, leading to poor interpretability of the results.

Based on theoretical results in Egozcue et al. (2006), who defines a Hilbert structure on a space of probability density functions on a compact interval (called a Bayes space), Delicado (2011) and Hron et al. (2014), propose a more reasonable approach to the problem of PCA for density functions. In particular, in Hron et al. (2014), the authors use the geometric properties of the Bayes space, coupled with a suitable transformation

from the Bayes space to an $L_2$ space, to perform PCA on a set of pdfs using FDA tools, and then map back the results to the Bayes space.

Another, perhaps less widely used, approach focuses on borrowing tools from symbolic data analysis (SDA) in the context of histogram data (Le-Rademacher and Billard, 2017; Nagabhushan and Pradeep Kumar, 2007; Rodríguez et al., 2000). Moreover, in Verde et al. (2015) some of these attempts are extended to generic distributional data using Wasserstein metrics.

Finally, Bigot et al. (2017) and Cazelles et al. (2018) propose two PCA formulations based on the geometric structure of the Wasserstein space: a *geodesic* PCA and a *log* PCA. In a similar fashion, the recent preprints of Chen et al. (2020) and Zhang et al. (2020) propose linear regression and autoregressive models, respectively, for distributional data using the Wasserstein geometry.

We now highlight some key aspects of the aforementioned approaches. Hron et al. (2014) assumes that all the probability measures have the same support. This is hardly verified in practice, so that to apply their techniques one needs either to truncate the support of some of the probability density functions, or to extend others (for instance, by adding a small constant value and renormalizing), leading to numerical instability as discussed in Sections 6.7 and 6.8.

The SDA-based methods in Le-Rademacher and Billard (2017); Nagabhushan and Pradeep Kumar (2007); Rodríguez et al. (2000) and Verde et al. (2015) share the poor interpretability of SDA.

The methods in Bigot et al. (2017), Cazelles et al. (2018), Chen et al. (2020) and Zhang et al. (2020) are based on the weak Riemannian structure of the Wasserstein space, cf. Section 6.2.2. Such structure enables the authors to borrow ideas and terminologies from statistical frameworks defined on Riemannian manifolds (see Banerjee et al., 2015; Bhattacharya et al., 2012; Fletcher, 2013; Huckemann et al., 2010b; Patrangenaru and Ellingson, 2015; Pennec, 2006, 2008). We can roughly distinguish those frameworks in two main approaches: the intrinsic/geodesic one and extrinsic/log one.

Briefly, intrinsic methods are defined using the metric structure of the Wasserstein space, working with geodesic curves and geodesic subsets, so that they faithfully respect the metric of the underlying space. However, in general, intrinsic methods present many practical difficulties in that the optimization problems they lead to are usually nontrivial, as we discuss in Section 6.5.3. Instances of intrinsic methods for distributional data are the *geodesic* PCA in Bigot et al. (2017) and, under some rather restrictive assumptions, the linear models in Chen et al. (2020) and the autoregressive models in Zhang et al. (2020), see Sections 6.3.3 and 6.3.4.

On the other hand, extrinsic methods resort to the linear structure of suitably defined tangent spaces, by mapping data from the Wasserstein space to the tangent (through the so-called *log* map) and then mapping back the results to the Wasserstein space (through the *exp* map). Of course, this approach is less respectful of the underlying geometry than the intrinsic one, but usually presents several numerical advantages. An example of such extrinsic methods defined in the Wasserstein space is the *log* PCA in Cazelles

et al. (2018).

The main issue with this *log* PCA is that the image of the *log* map inside the tangent of the Wasserstein space is not a linear space, but rather a convex cone embedded in a linear space (see Section 6.2.2). Hence, while exploiting the linear structure of the tangent, it is possible that the projection of some points onto the principal components end up outside of the cone. For these points, the *exp* map from the tangent to the Wasserstein space used in Cazelles et al. (2018) is not a metric projection, which in general is not available, so that the results in this setting are hardly interpretable.

### 6.1.2 Contributions and outline

The contribution of this work is three folded. First, we propose alternative PCA and regression models for distributional data in the Wasserstein space. We term these models *projected*, in opposition to the *log* PCA in Cazelles et al. (2018). Second, by exploiting a geometric characterization of Wasserstein space closely related to its weak Riemannian structure, we build a novel approximation of the Wasserstein space using monotone B-spline. This allows us to represent the space of probability measures as a convex polytope in $\mathbb{R}^J$. Lastly, we obtain faster optimization routines for the *geodesic* PCAs defined in Bigot et al. (2017), exploiting the aforementioned B-spline representation.

Our *projected* framework lies in between the *log* one and the *geodesic* one, since we use an analogous to the *log* map to transform our data, as for extrinsic methods, but do not resort to the *exp* map to return to the Wasserstein space, using instead the metric projection operator. Thanks to this, our *projected* methods are more respectful of the underlying geometry than the *log* ones, while at the same time retaining the same reduced computational complexity. Thus, the *projected* methods expand the range of situations where *extrinsic* methods are an effective and efficient alternative to intrinsic tools: in our examples, the performance loss in general is marginal (see Section 6.7).

By centering the analysis in appropriate points of the Wasserstein space, one can identify the space of probability measures (with finite second moment) with the space of square integrable monotonically non-decreasing functions on a compact set. We use a suitable quadratic B-spline expansion to get a very handy representation of such functions. Through such B-spline expansion, it is possible to approximate the metric projection onto the Wasserstein space as a constrained quadratic optimization problem over a convex polytope, that is a well-established problem, cf. Potra and Wright (2000). This allows us to exploit the underlying linear structure of an $L_2$ space, so that all the machinery developed for functional data analysis can be directly applied to this setting. We address the issue of interpretability of the results, tackling a number of diverse applications and developing different ways to measure the loss of information caused by the *extrinsic* nature of our methods.

We observe that the idea of representing nondecreasing functions through B-splines for statistical purposes has been proposed also by Das and Ghosal (2017), in the context of Bayesian quantile regression, where the authors use B-splines with (random)

monotonic coefficients as a generative model for random quantile functions. However, their focus is on defining a generative model, and not on developing a statistical setting exploiting the geometry given by the constrained representation. Along this direction, they do not restrict their attention to quadratic splines and consider cubic ones.

As already mentioned, a further contribution of this work is the derivation of alternative numerical optimization schemes for the *geodesic* PCA in Bigot et al. (2017) and Cazelles et al. (2018), based on the proposed quadratic B-spline expansion.

The remaining of the paper is organized as follows. Section 6.2 covers the basic concepts of Wasserstein distance and the weak Riemannian structure of the Wasserstein space, along with a brief discussion on a suitable way to exploit such structure for our purposes. Section 6.3 defines the *projected* PCA and *projected* regression in a general setting. In Section 6.4 we discuss the choice of the base point in which we center our analysis and how to efficiently approximate the metric projection through B-splines; in Section 6.5 we present the numerical algorithms needed to compute our *projected* methods and an alternative optimization routine for the *geodesic* PCA in Cazelles et al. (2018). Section 6.6 discusses the asymptotic properties of the spline approximation and of the *projected* models, establishing consistency of the estimators under some assumptions. Numerical illustrations on real and simulated data sets are shown in Sections 6.7 and 6.8. In particular, we apply our projected methods to two real world problems: we perform PCA on the US data on Covid-19 mortality by age and sex and perform a distribution regression to forecast the wind speed near a wind farm. Finally, the article concludes in Section 6.9. The Appendix collects all the proofs of the theoretical results, additional details on the simplicial PCA and regression, and further simulations. Code for reproducing the numerical results is available at `https://github.com/mberaha/ProjectedWasserstein`.

## 6.2 Preliminaries

In the following, we will consider probability measures on the real line $\mathbb{R}$ endowed with the usual Borel $\sigma$-field, we will skip references to the $\sigma$-field whenever it is obvious.

Given a measure $\mu$ on $\mathbb{R}$ define its cumulative distribution function $F_\mu(x) = \mu((-\infty, x])$ for $x \in \mathbb{R}$ and the associated quantile function $F_\mu^-(t) = \inf\{x \in \mathbb{R} : \ t \leqslant F_\mu(x)\}$. When $F_\mu$ is continuous and strictly monotonically increasing, $F_\mu^- = (F_\mu)^{-1}$.

### 6.2.1 Wasserstein metric and Wasserstein spaces

We start by recalling the definition of the 2-Wasserstein distance between two probability measures $\mu, \nu$ on $\mathbb{R}$:

$$W_2^2(\mu, \nu) = \inf_{\gamma \in \Gamma(\mu,\nu)} \int_{\mathbb{R}\times\mathbb{R}} |x - y|^2 d\gamma(x, y), \tag{6.1}$$

where $\Gamma(\mu, \nu)$ is the collection of all probability measures on $\mathbb{R} \times \mathbb{R}$ with marginals $\mu$ and $\nu$. Closely related to the definition of Wasserstein distance lies the one of Optimal Transport (OT). In particular, (6.1) identifies the Wasserstein distance with the minimal total transportation cost between $\mu$ and $\nu$ in the Kantorovich problem with quadratic cost (Ambrosio et al., 2008).

For our purposes, it is convenient to consider another formulation of the OT problem, originally introduced in Monge (1781). Given two measures $\mu, \nu$ as before, the optimal transport map from $\mu$ to $\nu$ is the solution of the problem

$$\inf_{T:T\#\mu=\nu} \int_{\Omega} |x - T(x)|^2 d\mu(x), \tag{6.2}$$

where $\#$ denotes the pushfoward operator, that is for any measurable set $B$ and measurable function

$$f : \mathbb{R} \to \mathbb{R}, \qquad (f\#\mu)(B) = \mu(f^{-1}(B)). \tag{6.3}$$

Note that any solution of (6.2) induces one and only one solution of (6.1); moreover if the OT problem has a unique solution, then also the Wasserstein distance problem has only one solution. However not all Wasserstein distance problems can be solved through Monge's formulation (Ambrosio et al., 2008).

The unidimensional setting is a remarkable exception in that there exist explicit formulas for both problems. In particular, the Wasserstein distance can be computed as

$$W_2^2(\mu, \nu) = \int_0^1 |F_\mu^-(s) - F_\nu^-(s)|^2 ds, \tag{6.4}$$

and, if the measure $\mu$ has no atoms, then there exists a unique solution to Monge's problem given by $T_\mu^\nu = F_\nu^- \circ F_\mu$. For a proof of these results, see Chapter 6 of Ambrosio et al. (2008).

It is clear that, in general, the Wasserstein distance between two probability measures can be unbounded (for instance when in (6.4) $F_\mu^-$ is not square integrable on $[0, 1]$). Nonetheless, when restricting the focus on the set of probability measures with finite second moment, then it holds that $W_2$ defines a metric (see, for instance, Chapter 7 of Villani, 2008). Formally, let the Wasserstein space:

$$\mathcal{W}_2(\mathbb{R}) = \left\{ \mu \in \mathcal{P}(\mathbb{R}) \ : \int_{\mathbb{R}} x^2 d\mu < +\infty \right\}$$

then $(\mathcal{W}_2(\mathbb{R}), W_2)$ is a separable complete metric space.

### 6.2.2 Weak Riemannian structure of the Wasserstein Space

Thanks to the uniqueness of the transport maps, by fixing an absolutely continuous (a.c.) probability measure $\mu \in \mathcal{W}_2(\mathbb{R})$, we can associate to any $\nu \in \mathcal{W}_2(\mathbb{R})$ the optimal transport map $T_\mu^\nu$. Since $\int_{\mathbb{R}} |T_\mu^\nu(x)|^2 d\mu = \int_{\mathbb{R}} x^2 d\nu$ we can define the following map $\varphi_\mu : \mathcal{W}_2(\mathbb{R}) \to L_2^\mu(\mathbb{R})$ with the rule: $\varphi_\mu(\nu) = T_\mu^\nu$.

We note several immediate but interesting properties of the map $\varphi_\mu$. First, it is an isometry (and so a homeomorphism onto its image) since

$$\int_{\mathbb{R}} |T_\mu^\nu(x) - T_\mu^\eta(x)|^2 d\mu = \int_{[0,1]} |F_\nu^- - F_\eta^-|^2 ds = W_2^2(\nu, \eta).$$

Second, the image of $\varphi_\mu$ is a closed convex cone in $L_2^\mu(\mathbb{R})$: a set closed under addition and positive scalar multiplication. In fact, for any $\lambda \geqslant 0$, $\lambda T_\mu^\nu$ is still a transport map from $\mu$ to another measure whose quantile is $\lambda F_\mu^-$; and similarly $T_\mu^\nu + T_\mu^\eta = (F_\nu^- + F_\eta^-) \circ F_\mu$. Being $\mathcal{W}_2(\mathbb{R})$ complete, $\varphi_\mu(\mathcal{W}_2(\mathbb{R}))$ is closed in $L_2^\mu(\mathbb{R})$. Third, $\varphi_\mu(\mu) = id_{\mathbb{R}}$ (where $id_C$ denotes the identity map of the set $C$). Finally, as shown in Panaretos and Zemel (2020), $\varphi_\mu$ is not surjective and $\varphi_\mu(\mathcal{W}_2(\mathbb{R}))$ is the set of $\mu$-a.e. non decreasing functions in $L_2^\mu(\mathbb{R})$.

The inverse of the map of $\varphi_\mu$ is the measure pushforward (see Equation 6.3) and it is defined on the whole $L_2^\mu(\mathbb{R})$: given $f \in L_2^\mu(\mathbb{R})$, then $\nu = f\#\mu$ is a measure in $\mathcal{W}_2(\mathbb{R})$. In fact:

$$\int |x|^2 d\nu = \int |f(x)|^2 d\mu = \|f\|_\mu^2$$

A natural way to define a tangent structure for $\mathcal{W}_2(\mathbb{R})$ is therefore to take advantage of the cone structure given by $\varphi_\mu$. In fact for closed convex cones, there are already notions of tangent cones. Similarly to Rockafellar and Wets (1998), Theorem 6.9, we can define:

$$\text{Tan}_\mu(\mathcal{W}_2(\mathbb{R})) := \text{Tan}_{id_{\mathbb{R}}}(L_2^\mu(\mathbb{R})) = \overline{\{f \in L_2^\mu(\mathbb{R}) | \exists h > 0 : id + hf \in \varphi_\mu(\mathcal{W}_2(\mathbb{R}))\}}^{L_2^\mu(\mathbb{R})} \tag{6.5}$$

We remark that Theorem 6.9 in Rockafellar and Wets (1998) is stated in $\mathbb{R}^n$, but it holds also more generally, for instance in an Hilbert space (see Aubin and Frankowska (2009), Chapter 4).

A geometric interpretation of (6.5) is the following. The tangent space consists of all the vectors $f$ that move the base point inside the cone $\varphi_\mu(\mathcal{W}_2(\mathbb{R}))$, when considered up to a scale factor $h$. Hence, $f$ plays the role of direction of a tangent vector going out from the tangent point. Furthermore, since for every $f \in \varphi_\mu(\mathcal{W}_2(\mathbb{R}))$ then $f + id \in \varphi_\mu(\mathcal{W}_2(\mathbb{R}))$ we have that $\varphi_\mu(\mathcal{W}_2(\mathbb{R}))$ is included in the tangent space. As shown later in this Section, the inclusion is strict and the tangent space is much larger than $\varphi_\mu(\mathcal{W}_2(\mathbb{R}))$.

Note that we can recover the definition of tangent space given by Ambrosio et al. (2008) and Panaretos and Zemel (2020) by a simple "change of variable": calling $g = id + hf$ then substituting $(g - id)/h$ in (6.5) gives the following definition of tangent

$$\text{Tan}_\mu(\mathcal{W}_2(\mathbb{R})) = \overline{\{\lambda(f - id) | f \in \varphi_\mu(\mathcal{W}_2(\mathbb{R})); \lambda > 0\}}^{L_2^\mu(\mathbb{R})},$$

which is the one given in Ambrosio et al. (2008) and Panaretos and Zemel (2020). As shown in Panaretos and Zemel (2020) the tangent cone $\text{Tan}_\mu(\mathcal{W}_2(\mathbb{R}))$ is indeed a linear space. For this reason we refer to it as tangent space, instead of cone.

In analogy to Riemannian geometry, following Ambrosio et al. (2008) and Panaretos and Zemel (2020), we define the $\log_\mu$ and $\exp_\mu$ maps. Having fixed $\mu$ absolutely continuous:

$$\begin{aligned} \log_\mu : \mathcal{W}_2(\mathbb{R}) &\to \text{Tan}_\mu(\mathcal{W}_2(\mathbb{R})) \\ \nu &\mapsto T_\mu^\nu - id \end{aligned} \qquad \begin{aligned} \exp_\mu : \text{Tan}_\mu(\mathcal{W}_2(\mathbb{R})) &\to \mathcal{W}_2(\mathbb{R}) \\ f &\mapsto (id + f)\#\mu \end{aligned} \tag{6.6}$$

We briefly highlight some properties of these maps; properties which immediately follows from the discussion above.

**Remark 21.** *The map $\log_\mu$ is defined on the whole space $\mathcal{W}_2(\mathbb{R})$. Moreover, it is clearly an isometry: $W_2(\eta, \nu) = \|\log_\mu(\eta) - \log_\mu(\nu)\|_{L_2^\mu(\mathbb{R})}$ (Panaretos and Zemel, 2020). This shows that there is no local-approximation issue when working in the tangent space, in contrast with the usual Riemannian manifold setting. There, the tangent space usually provides good approximation only in a neighborhood of the tangent point.*

**Remark 22.** *The map $\log_\mu$ is not surjective on Tan$_\mu$, indeed its image Im$(\log_\mu)$ is a closed convex subset of $L_2^\mu(\mathbb{R})$ given by all the maps $f$ such that $f + id \in \varphi_\mu(\mathcal{W}_2(\mathbb{R}))$, that is, $f + id$ is $\mu$-a.e. increasing. The restriction of $\exp_\mu$ on Im$(\log_\mu)$, henceforth denoted by $\exp_{\mu|\log_\mu(\mathcal{W}_2(\mathbb{R}))}$, is an isometric homeomorphism and its inverse is $\log_\mu$. In particular, we observe that $\log_\mu \circ \exp_\mu$ is not a metric projection in $L_2^\mu$. That is, in general $\log_\mu \circ \exp_\mu(f) \neq \arg\min_{g \in Im(\log_\mu)} ||f - g||_{L_2^\mu}$.*

### 6.2.3 Intrinsic and extrinsic methods in the Wasserstein space

As mentioned in Section 6.1.1, borrowing ideas from Riemannian geometry leads to discerning statistical methods on the Wasserstein space in the classes of *intrinsic* and *extrinsic* methods.

The Weak Riemannian structure presented in Section 6.2.2 provides a suitable environment for developing intrinsic methods. In fact, the geodesic structure of $\mathcal{W}_2(\mathbb{R})$ can be recovered through the linear structure of any $L_2^\mu(\mathbb{R})$ space through the isometry $\varphi_\mu$. Pointwise interpolation of the transport maps coincide with the geodesic between measures. In other words, given $\mu$ a.c., the geodesic between $\nu$ and $\eta$ is given by:

$$\gamma(t) = ((1-t) \cdot T_\mu^\nu + t \cdot T_\mu^\eta)\#\mu \tag{6.7}$$

Thus, such geodesic structure can be recovered in many different (but equivalent) ways, depending on $\mu$.

On the other hand, Remark 21 motivates the development of extrinsic tools, since working in the image of $\log_\mu$ inside the tangent space Tan$_\mu$ is exactly like working in $\mathcal{W}_2(\mathbb{R})$. This is not common in Riemannian manifold framework, since usually the tangent space provides a good approximation only near to the tangent point. As a consequence, if in the general Riemannian manifold framework the choice of the tangent point $\mu$ is crucial (since results for extrinsic methods might be significantly altered for different choices of $\mu$) when working with $\mathcal{W}_2(\mathbb{R})$ this is not the case.

To further motivate this key point, consider $\mu$ and $\nu$ a.c. measures; the maps $\log_\nu \circ (\exp_{\mu|\log_\mu(\mathcal{W}_2(\mathbb{R}))})$ and $\varphi_\nu \circ \varphi_\mu^{-1}$ are isometric homeomorphisms (as composition of isometries and homeomorphisms). In other words, they preserve distances and send border elements of $\log_\mu(\mathcal{W}_2(\mathbb{R}))$ or $\varphi_\mu(\mathcal{W}_2(\mathbb{R}))$ into border elements of $\log_\nu(\mathcal{W}_2(\mathbb{R}))$ and $\varphi_\nu(\mathcal{W}_2(\mathbb{R}))$, respectively, and the same with internal points (and so in particular, they preserve distances from any point to the border). In Chen et al. (2020), Bigot et al. (2017) and Zhang et al. (2020) $\mu$ is chosen as the barycentric measure $\bar{x}$ of the observations $x_i \in \mathcal{W}_2(\mathbb{R})$. The discussion above implies that considering the tangent space at the Wasserstein barycenter $\bar{x}$ and working on $\log_{\bar{x}}(x_i) = \log_{\bar{x}}(x_i) - \log_{\bar{x}}(\bar{x})$ is exactly the same as considering the tangent space at any $\mu$ a.c. and working on $\log_\mu(x_i) - \log_\mu(\bar{x})$ for our statistical purposes. So the choice of the tangent space from the theoretical point of view is completely arbitrary.

Moreover, centering the analysis in the barycenter presents a drawback when studying asymptotic properties of the models under consideration, since $\bar{x}$ changes as the sample size grows. In Section 6.4.1 we propose to fix $\mu$ as the uniform measure on $[0, 1]$. This choice not only allows us to derive empirical methods that are extremely simple to implement, cf. Section 6.5, but also allows us to study asymptotic properties of the models in Section 6.6.2 without resorting to parallel transport, as done for instance in Chen et al. (2020).

### 6.2.4 Tangent vs. $L_2^\mu$

Lastly, we briefly discuss the major differences between using a tangent space representation of $\mathcal{W}_2(\mathbb{R})$ and using the representation given by some $\varphi_\mu$.

We recall that, for a fixed $\mu$ a.c., the two representations are indeed quite similar $\varphi_\mu(\nu) = T_\mu^\nu$, $\log_\mu(\nu) = T_\mu^\nu - id$; a priori one may prefer the tangent representation, because it already expresses data as vectors coming out of a point. Therefore, for instance, it might result practically more convenient to center the analysis in the barycenter and work on vectors, taking away any "data centering" issues. At the same time, also notational coherence with already existing methods might benefit from this choice.

However, especially when dealing with extrinsic techniques, we found slightly more practical to use the $\varphi_\mu$ representation in that it is more straightforward to represent $\varphi_\mu(\mathcal{W}_2(\mathbb{R}))$ compared to $\log_\mu(\mathcal{W}_2(\mathbb{R}))$: the first one can in fact be represented directly as the cone of the $\mu$-a.e non-decreasing functions.

## 6.3 Projected Models in the Wasserstein Space

In this section, exploiting the embeddings given by $\varphi_\mu$, we define a class of *projected* statistical methods to perform extrinsic analysis for data in the Wasserstein space.

To give a general framework, we do not restrict our attention to a particular $\varphi_\mu$ yet, even though in Section 6.4 we argue that a natural choice which allows an easier

implementation of the empirical methods is letting $\mu$ be the uniform distribution on $[0, 1]$. Hence, for the sake of notation, we consider a generic case of data lying in a closed convex cone $X$ inside a separable Hilbert space $H$. In our setting, $H$ would be $L_2^\mu(\mathbb{R})$ and $X = \varphi_\mu(\mathcal{W}_2(\mathbb{R}))$, for some $\mu \in \mathcal{W}_2(\mathbb{R})$ absolutely continuous.

### 6.3.1 Principal component analysis

We start by defining one of the main contributions of our work: the *projected* PCA. We recall that for an $H$-valued random variable $\mathcal{X}$, PCA is a well established technique and amounts to finding the eigenfunctions of the Karhunen-Loéve expansion of the covariance operator of $\mathcal{X}$, see Ramsay (2004). Observe that any $X$-valued random variable can be considered as an $H$-valued one (by the inclusion map), so that a notion of PCA is already available.

When defining principal components, a key notion is the one of dimension of the principal component (PC). In this work, principal components will be closed convex subsets of $H$, and we will always define the dimension of a subset of $H$ as the dimension of the smallest affine subset of $H$ containing it. For a generic closed convex set $C \subset H$, let $\Pi_C$ denote the metric projection onto $C$: $\Pi_C(x) := \arg\min_{c \in C} ||x - c||$ and, for a set of vectors $U$, denote with $Sp(U)$ its linear span.

In what follows, we denote by $x_0$ the "center" of the PCA. For us, $x_0 = \mathbb{E}[\mathcal{X}]$, or its empirical counterpart. To have a well defined PCA, we always assume that $x_0$ belongs to the relative interior of the convex hull of the support of $\mathcal{X}$, see Appendix 6.11 for the definition of relative interior and further details. This is a rather technical hypothesis but it is not a restrictive one. For instance, it is always verified for empirical measures and when $X \subseteq \mathbb{R}^d$ and hence for our empirical methods, cf. Section 6.5.1.

**Definition 35.** *(Projected PCA). Given $\mathcal{X}$ a random variable with values in $X \subset H$, let $U_k = \{w_1, ..., w_k\}$ be its first $k$ $H$-principal components centered in $x_0 = \mathbb{E}[\mathcal{X}]$. A $(k, x_0)-$projected principal component of $\mathcal{X}$ is the biggest closed convex subset $U_X^{x_0,k}$ of $X$ such that:*

1. $x_0 \in U_X^{x_0,k}$,
2. $dim(U_X^{x_0,k}) = k$, *and*
3. $U_X^{x_0,k} \subseteq \Pi_X(Sp(U_k))$.

In other words, the projected principal component is obtained by approximating the span of the principal components found in $H$, with convex subsets in $X$. Note that the principal components in $H$ might "capture" some variability which is not present when measuring distances inside $X$. In fact the projection of a point belonging to $X$ onto a direction $w_j$ might end up being outside $X$, see Section 6.3.3. However, as we will show in Section 6.7, in our examples the projected PCA behaves well and this issue does not seem to affect significantly the performance.

**Remark 23.** *Convex sets are essential in our analysis since, thanks to* (6.7) *convex sets in $X$ are precisely the subsets of $\mathcal{W}_2(\mathbb{R})$ which are geodesically complete: the geodesic connecting any pair of points in the subset, is contained in the subset. Geodesic subsets are a natural generalization of linear spaces.*

**Remark 24.** *The metric projection of a linear subspace onto a convex subset can end up being a nonconvex set. In addition to that, while loosing convexity, the dimension of the metric projection of a convex subset can be bigger of the dimension of the original subset. A simple example where both cases happen is the projection of $y = -x$ onto $x, y \geqslant 0$ in $\mathbb{R}^2$.*

We observe that inside a projected principal component, we have a preferential orthonormal basis given by the principal components in $H$; for this reason we call $U_k = \{w_1, ..., w_k\}$ *principal directions*.

Although it might seem impractical to find the projected component, the following Lemma provides a more convenient alternative characterization.

**Lemma 12.** *Let $x_0$ and $U_X^{x_0,k}$ be as in Definition 35, then $U_X^{x_0,k} = (x_0 + Sp(U_k)) \cap X$.*

Natural alternatives to Definition 35 would be, for instance, to let the projected principal directions (component) be the metric projection of $w_1, \dots, w_k$ (the linear span of $\{w_1, \dots, w_k\}$) onto $X$, respectively. In the former case, the projection would not guarantee the orthogonality of the projected directions, which is instead essential to properly explore the variability. Moreover, since the "tip" of the projected unit vectors would likely lie on the border of $X$, the projection of a new observations on a direction would still lie outside of $X$ as soon as the score associated to that direction is larger than $1$. The latter case, instead, presents the drawbacks pointed out in Remark 24.

We argue that, despite its simplicity, Definition 35 is indeed very well suited for statistical analysis in the Wasserstein Space. For instance, we are guaranteed that, as the dimension grows up, the $k$ projected components provide a monotonically better fit to the data. This is easily verified because $\Pi_X$ is a strictly non-expansive operator, being $X$ closed and convex (see Deutsch (2012)), which implies the following Proposition.

**Proposition 29.** *With the same notation as Definition 35, for any $x \in X$ we have:*

$$\|\Pi_{U_X^{x_0,k}}(x) - x\| \geqslant \|\Pi_{U_X^{x_0,k+1}}(x) - x\| \to 0 \text{ with } k \to +\infty.$$

Once a principal component is found, a classical task that one may want to perform is to project a new "observation" $x^* \in X$ onto $U_X^{x_0,k}$, for instance for dimensionality reduction purposes. In general, the metric projection on generic convex subsets might be arduous to find, we will deal with this issue in Section 6.4. Nevertheless, we can use the following Proposition to reduce in advance the dimension of the parameters involved in the problem; turning it into a projection problem inside the principal projected component, which allows for faster computations (see Equation 6.13).

**Proposition 30.** *Let $x^* \in X$ and let $\Pi_k$ be the orthogonal projection on $Span(U_k)$. The projection of $x^*$ onto $U_X^{x_0,k}$ is given by*

$$\underset{v' \in U_X^{x_0,k}}{\arg\min} \|x^* - v'\| = \Pi_{Sp(U_k) \cap (X - x_0)}(\Pi_k(x^* - x_0)) + x_0. \tag{6.8}$$

Lastly, we observe that, since projected principal components are not linear subspaces, the scores of some points on a principal direction can vary as we increase the dimension of the principal component.

### 6.3.2 Regression

Broadly speaking, a regression model between two variables with values in two different spaces is given by an operator between such spaces, which for every input value of the independent variable, returns a predicted value for the dependent variable. In the following, let us denote with $\mathcal{Z}$ the independent variable and with $\mathcal{Y}$ the dependent one. A regression model is usually understood as an operator $\Gamma$ specifying the conditional value of $\mathcal{Y}$ given $\mathcal{Z}$, that is, $\mathbb{E}[\mathcal{Y}|\mathcal{Z}] = \Gamma(\mathcal{Z})$.

If the spaces where $\mathcal{Z}$ and $\mathcal{Y}$ take values possess a linear structure, this linearity is usually exploited by means of a (kernel) linear operator, with possibly an "intercept" term. To define our *projected* regression model, we want to exploit the cone structure of $X$ in a similar fashion. In fact, such linear kernel operators combine good optimization properties and interpretability since their kernels can provide insights into the analysis, much like coefficients in multivariate linear regression.

We treat separately the cases where the $X$-valued variable is the independent or the dependent one. The case when both variables are $X$-valued follows naturally. To keep the notation light, in what follows we will not distinguish between "proper" linear operators and linear operators with an added intercept term, which could as well be employed in all the incoming definitions to gain flexibility.

Consider the case in which we have an independent $X$-valued random variable, and denote with $V$ the space where the dependent variable takes value. Despite the fact that $X$ is not a linear space, with an abuse of notation, we call "linear" an operator which respect sum and positive scalar multiplication for elements in $X$. Such operators are in fact obtained by restricting on $X$ linear operators defined on $H$. Following this idea, in order to define linear regression for an $X$-valued independent random variable, we consider such variable as $H$-valued, obtain the regression operator and then take the restriction of the operator on $X$. In this way, when $H = L_2^\mu(\mathbb{R})$ and $X = \varphi_\mu(\mathcal{W}_2(\mathbb{R}))$, it is possible to exploit the classical FDA framework to perform all kinds of distribution on scalar/vector/etc... regression. For brevity, we report only the definition with $V = \mathbb{R}$.

**Definition 36.** *Let $\mathcal{Z}$ an $X$-valued random variable, and $\mathcal{Y}$ a real valued one. Let $\Gamma_\beta : H \to \mathbb{R}$ be a functional linear regression model for such variables, with $\mathcal{Z}$ considered as $H$-valued and $\Gamma_\beta(v) = \langle \beta, v \rangle$. A projected linear regression model for $(\mathcal{Z}, \mathcal{Y})$ is given by $(\Gamma_\beta)_{|X}$.*

Now we turn to the cases which feature an $X$ valued independent variable and a $Z$ valued dependent one, for $Z$ a generic Hilbert space. Through the inclusion $X \hookrightarrow H$, we can consider a regression problem with $X$-valued dependent variable, as a problem with $H$-valued dependent variable. Comparing this situation with the previous one, it is clear that we now face a "dual" problem. Indeed, while before we needed to restrict the domain from $H$ to $X$, we now need to force the codomain of $\Gamma$ to lie inside $X$. We would like to retain the same properties that make linear kernel operators appealing as regression operators between Hilbert spaces. A possibility could be considering a linear kernel operator $\Gamma$ with values in $H$ and restricting it to $\Gamma^{-1}(X)$. However, this would imply that for any $z \notin \Gamma^{-1}(X)$ no prediction would be available.

We argue that a more reasonable approach consists in finding an operator $\Gamma_{\mathrm{P}} : Z \to X$ as close as possible (in some sense that will be clear later) to the linear kernel operator $\Gamma$ aforementioned. Hence, we relax the linearity assumption in favor of Lipschitzianity, and take as regression operator $\Pi_X \circ \Gamma$, whose image always lies in $X$. Note that $\Gamma_{\mathrm{P}}$ inherits the interpretability of the kernel of $\Gamma$.

To motivate such choice, we give the following notion of a projected operator.

**Definition 37.** *Let $Z$ be a normed space and consider $\mathcal{Z}$ a $Z$-valued random variable. Let $\Gamma : Z \to H$ a generic Lipschitz operator between $Z$ and $H$. A $(\mathcal{Z}, X)$-projection of $\Gamma$ is an operator $\Gamma_{\mathrm{P}} : Z \to X$ such that:*

$$\Gamma_{\mathrm{P}} = \operatorname*{arg\,min}_{T:Z\to X} \mathbb{E}_{\mathcal{Z}}[\|\Gamma(v) - T(v)\|^2]$$

In other words, $\Gamma_{\mathrm{P}}$ provides the best pointwise approximation of the $H$-valued operator $\Gamma$, averaged w.r.t. the measure induced by $\mathcal{Z}$. Hence, given a $\mathcal{Z}$ a $Z$-valued random variable and $\mathcal{Y}$ an $X$-valued random variable and a linear regression model $\Gamma : Z \to H$ for $(\mathcal{Z}, \mathcal{Y})$, the projected regression model induced by $\Gamma$ is $\Gamma_{\mathrm{P}}$.

**Proposition 31.** *With the same notation as above, if $\mathbb{E}\left[\|\mathcal{Z}\|^2\right] < \infty$, then $\Gamma_{\mathrm{P}} = \Pi_X \circ \Gamma$.*

*Proof.* For any $T : Z \to X$, it holds: $\|\Gamma(z) - \Pi_X(\Gamma(z))\| \leqslant \|\Gamma(v) - T(v)\|$. Moreover, $\Gamma$ and $\Pi_X \circ \Gamma$ are Lipschitz, and being $\Pi_X$ non-expansive, they share the same constant $L > 0$:

$$\|\Gamma(v) - \Pi_X \circ \Gamma(v)\|^2 \leqslant 2L\|v\|^2$$

and thus $\mathbb{E}_{\mathcal{Z}}[\|\Gamma(z) - \Pi_X \circ \Gamma(z)\|^2]$ is bounded iff $\mathcal{Z}$ has finite second moment. □

The only case left out from the treatment above is when both the independent and the dependent variables are X-valued. This case, however, follows naturally by combining the two approaches and we report the definition below.

**Definition 38.** *Let $\mathcal{Z}$ and $\mathcal{Y}$ two $X$-valued random variables. Let $\Gamma : H \to H$ be a functional linear regression model for the variables considered as $H$-valued. A projected linear regression model for $(\mathcal{Z}, \mathcal{Y})$ is given by $(\Pi_X \circ \Gamma)_{|X}$.*

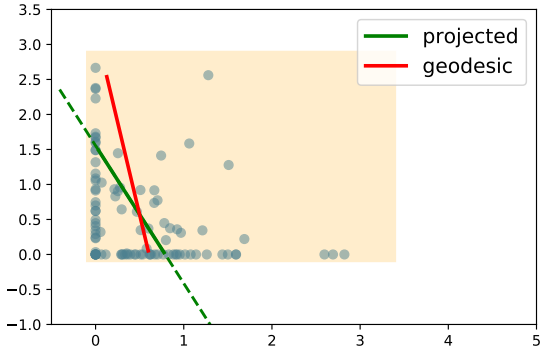



**Figure 6.1:** *Comparison of projected and geodesic PCA when $H = \mathbb{R}^2$ and $X$ is the shaded rectangle. The projected principal direction is rather different from the geodesic one because most of the observations (blue dots) are concentrated around the borders*

**Remark 25.** *When considering a regression with $X$-valued independent variable, one may want to relax the restriction on $X$ in Definition 36 for various reasons; for instance one may have measurement errors, or by design the test set may consider points also outside $X$. In such cases it is worth considering the problem of how many continuous linear extensions of $\Gamma_{|X}$ are possible on the whole $H$. A sufficient condition for the uniqueness of such extension is the following: there exist a sequence of linear subspaces of $H$, say $\{H_J\}_{J\geqslant 1}$, such that $\bigcup_J H_J$ is dense in $H$ and $X_J := H_J \cap X$ contains a basis of $H_J$ for every $J$.*

**Remark 26.** *When $H = L_2^\mu(\mathbb{R})$ and $X = \varphi_\mu(\mathcal{W}_2(\mathbb{R}))$ the condition in Remark 25 is verified, for instance, by Remark 28 in Section 6.4.3. Moreover, observe that the uniqueness of the extension can also be proven thank to Jordan's representation of functions $f : \mathbb{R} \to \mathbb{R}$ with bounded variation (BV). In fact any $f$ with BV can be written as the difference of monotone functions and thus $\Gamma(f)$ is fixed. Then by the density of BV functions in $H$, we define $\Gamma$ on the remaining elements of $H$.*

### 6.3.3 Comparison with intrinsic methods

We now compare the projected methods defined earlier in this Section and the intrinsic counterparts. In particular, we focus on the *geodesic* PCA defined in Bigot et al. (2017) and Cazelles et al. (2018) and on the distribution on distribution regression model in Chen et al. (2020).

Bigot et al. (2017) and Cazelles et al. (2018) define two different PCA, namely a global and a nested one; in particular the nested approach presents analogies with other PCAs developed for manifold valued random variables (Huckemann and Eltzner, 2018; Jung et al., 2012; Pennec, 2018); we report the two definitions below.

**Definition 39.** *(Global geodesic PCA) Let $\mathcal{X}$ a random variable with values in $X$ with $\mathbb{E}[\mathcal{X}] = x_0$. A $(k, x_0)$-global geodesic PC is a set $C^*$ minimizing $\mathbb{E}\left[d(\mathcal{X}, C)^2\right]$ over the*

*closed convex sets $C \subset X$ such that $x_0 \in C$ and $dim(C) \leqslant k$*

**Definition 40.** *(Nested geodesic PCA) Let $\mathcal{X}$ a random variable with values in $X$ with $\mathbb{E}[\mathcal{X}] = x_0$. For $k = 1$, a $(k, x_0)$-nested geodesic PC is a set $C_k^*$ such that $C_k^*$ is a minimizer of $\mathbb{E}\left[d(\mathcal{X}, C)^2\right]$ over the closed convex sets $C \subset X$ such that $x_0 \in C$ and $dim(C) \leqslant k$; for $k \geqslant 1$, a $(k, x_0)$-nested geodesic PC is a set $C_k^*$ such that $C_k^*$ is a minimizer of $\mathbb{E}\left[d(\mathcal{X}, C)^2\right]$ over the closed convex sets $C \subset X$ such that: $x_0 \in C$, $dim(C) \leqslant k$, and $C \supset C_{k-1}^*$, where $C_{k-1}^*$ is a $(k-1, x_0)$-nested geodesic PC.*

The first key difference between the global and the nested geodesic PCA is that the latter provides a notion of preferential directions in the principal component, while the first one does not. In fact, the first nested principal component corresponds to the first principal direction, and it is possible to find the remaining principal directions by imposing orthogonality constraints as we obtain nested PCs of higher dimensions. Thus, the nested geodesic PCA is more suitable to explore and visualize the variability in a data set, see also Section 6.7. On the other hand, exactly because of the lack of such constraints, the global PCA is in general more flexible and provides superior performance in terms of *reconstruction error*, cf. Section 6.7.

Comparing these definitions with the one of our projected PCA, the key difference is that geodesic PCAs do not exploit the Hilbert structure of $H$. Thus, as we discuss in Section 6.5.3, the numerical routines needed to find such principal components rely on nonlinear constrained optimization, which can be extremely demanding and nontrivial to implement. This is in sharp contrast with our projected PCA in Definition 35, that, thanks to Lemma 12 can be straightforwardly computed. However, as a result, the projected PCA is in general less respectful of the underlying metric structure. By investigating this issue in simpler settings, for instance when $H = \mathbb{R}^d$ and $X$ is a convex polytope in $\mathbb{R}^d$, we noticed that the differences between the projected principal directions and the nested geodesic ones become appreciable only if the random variable $\mathcal{X}$ gives significant probability to values near the borders of $X$. See for instance Figure 6.1. While this intuition remains valid also in the more complex setting that we investigate in this paper, it is harder to imagine realizations of $\mathcal{X}$ near the borders of $X$.

Note that the interpretability of the projected PCA is determined by the level of discrepancy between the definitions, as in Figure 6.1, which depends on how much variability it is correctly captured by the component, that is how much of the variability captured by the projected component lies in $X$. This intuition is formalized in Section 6.7.2 where two measures of "reliability" of the projected PCA are proposed.

Turning to the regression context, Chen et al. (2020) define a distribution on distribution linear regression model in the Wasserstein space. Their approach considers two different tangent spaces of $\mathcal{W}_2(\mathbb{R})$ (the first one centered in the barycenter of the independent variable and the second one centered in the barycenter of the dependent variable) and map the observations to the corresponding tangent spaces. They then use FDA tools to estimate a functional linear model $\widehat{\Gamma}$ between those two spaces. When the image of the regression operator $\Gamma$ lies inside the image of the log map centered

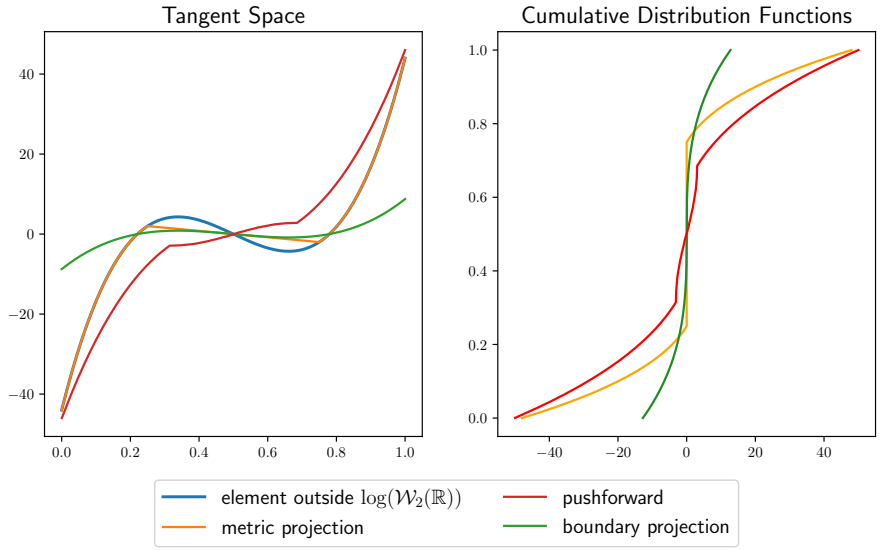



**Figure 6.2:** *Comparison between different projections onto $X$ for a point $x \in H \backslash X$ (blue line) in the tangent space (left panel) and the associated cumulative distribution functions (right panel) when the base point $\mu$ is the uniform measure on $[0, 1]$. The orange, green and red curves are obtained with metric projection, boundary projection and $\log_\mu \circ \exp_\mu$ respectively.*

in the dependent variable's barycenter, their distribution on distribution regression can be considered a properly intrinsic method. This assumption is used to prove asymptotic properties of their methodology, but as the authors in Chen et al. (2020) notice, is hardly verified in practice, so that whenever the output of the regression operator is not a distribution, they resort to squeezing such a value with some scalar multiplication, namely "boundary projection", which in general is not a metric projection. The boundary projection step gives an extrinsic nature to their model and we provide further comparisons with our methods in Section 6.3.4.

### 6.3.4 Comparison with other extrinsic methods

In this section, we offer a comparison of our projected methods with other extrinsic methods, namely the log PCA in Cazelles et al. (2018) and the distribution on distribution regression in Chen et al. (2020), which, as outlined in the previous section, may behave as an extrinsic method. Let us start with the former.

Cazelles et al. (2018) propose the definition of a log PCA as an alternative to the geodesic PCAs in Bigot et al. (2017). Both the log and the projected PCA are extrinsic methods: they proceed by carrying out the PCA in a linear space $H$ and then map back the results to the Wasserstein space, following an approach which had already been proposed by Fletcher et al. (2004).

For the log PCA, $H$ is the tangent space at $\mu$, for the projected $H$ is $L_2^\mu(\mathbb{R})$. Given $U_k = \{w_1, \ldots, w_k\}$ the first $k$ $H$-principal components, the log principal component in $\mathcal{W}_2(\mathbb{R})$ is $\exp_\mu(Sp(U_k))$ . Analogously, by considering the convex cone $X =: \log_\mu(\mathcal{W}_2(\mathbb{R})) \subseteq H$, the principal component in $X$ is $\log_\mu\left(\exp_\mu(Sp(U_k))\right)$.

We notice two key differences between the log and projected PCA. First, as pointed out in Remark 22, $\log_\mu \circ \exp_\mu$ is not a metric projection in $L_2^\mu$ so that given a point $x \in H \backslash X$, $\log_\mu(\exp_\mu(x))$ might end up being extremely different from $x$. See for instance Figure 6.2 where for a point $x$ (blue line) that is close (in the $L_2^\mu$ norm) to $X$, $\log_\mu(\exp_\mu(x))$ turns out to be quite far from $x$. In the context of PCA, this means that as soon as the projection onto $Sp(U_k)$ of observation lies outside of $X$, the log PCA quickly loses its interpretability. Second, as discussed in Remark 24, there is no guarantee that $\log_\mu\left(\exp_\mu(Sp(U_k))\right)$ is contained in $Sp(U_k)$, its dimension might increase and it might not even be convex. For this same reason, in general, log PCA cannot define a set of (orthogonal) principal directions which span the principal component. Hence, it is not possible to work directly on the scores of the PCA.

Combined, we believe that the above mentioned issues present a major drawback of the log PCA when compared to the projected PCA, as they prevent the possibility of doing proper dimensionality reduction and working on the scores of data points on the principal components. Finally, we also point out that approximating the $\exp_\mu$ map is a nontrivial task, involving computing numerically the preimages of an arbitrary large number of sets and numerical differentiation, that can lead to numerical instability of the log PCA.

We end this disccussion with a comparison between the boundary projection in Chen et al. (2020) and the metric projection. Their difference, for a possible regression output $x \in H \backslash X$ is depicted in Figure 6.2. Note that, by construction, such a procedure shrinks the tails of the output. Even when the regression output is slightly outside the image of the log map, the boundary projection result can be extremely far from the regression output and from the metric projection in terms of Wasserstein distance. For example, in Figure 6.2, the regression output and the projected method assign positive probability to values in the range $[-45, 45]$, while the output of the boundary projection assigns zero probability to values outside $[-17, 17]$. This underrepresentation of the variability might be a crucial issue depending on the application considered.

## 6.4 Computing the metric projection through B-spline approximation

The projected methods defined in Section 6.3 depend heavily on the availability of projection operators on the closed convex cone $X = \varphi_\mu(\mathcal{W}_2(\mathbb{R}))$. Being $X$ a cone inside a linear space, such operators are always well defined, but their implementation might be nontrivial. In this Section, we present a possible solution to this problem, based on choosing a particular $\mu$ as base point and constructing a B-spline representation of the

cone $X$.

### 6.4.1 Choosing $\mu$ as the uniform distribution on $[0, 1]$

As already mentioned, our projected methods can be carried out by choosing $\mu$ arbitrarily and there is no theoretical difference between different choices of $\mu$, cf. Section 6.2.2. Nonetheless, in practice, a clever choice of $\mu$ can lead to substantially easier and more numerically stable algorithms. For instance, by choosing a measure $\mu$ with compact support $C$ in $\mathbb{R}$, then the ambient space becomes $L_2^\mu(C)$ since we work up to zero-measure sets. This greatly simplifies any numerical procedure since we could work with grids over bounded sets, and do not need to resort to any truncation procedure, which would be mandatory in case the support of $\mu$ was unbounded. Moreover, note that evaluating the maps $\varphi_\mu$ in a certain measure $\nu$ amounts to computing the transport map $T_\mu^\nu = F_\nu^- \circ F_\mu$, hence it is clear that the choice of $F_\mu$ numerically influences the results.

For the aforementioned reasons, we argue that a reasonable choice is to center our analysis in $\mu = U([0, 1])$. In fact, in this case, $L_2^\mu(\mathbb{R}) = L_2([0, 1])$, and $F_\mu = id_{[0,1]}$ (the transport maps are simply given by quantile functions).

### 6.4.2 Metric Projection

Having chosen $\mu$ as Section 6.4.1 leads to an explicit characterization of the image of $\varphi_\mu$ as the set of square integrable a.e. non-decreasing functions on $[0, 1]$. Hence, the operator $\Pi_X$ in Section 6.3 is the metric projection onto the cone of a.e. non-decreasing functions in $L_2([0, 1])$.

Projection onto monotone functions has been widely studied in the field of *order restricted* inference, (Anevski et al., 2006; Dykstra et al., 2012). For instance, in Anevski and Soulier (2011) an explicit characterization of such a projection is given, which however does not lead to a closed form solution, while in Ayer et al. (1955) several numerical algorithms to approximate the projection operator are proposed. Those algorithms are based on approximating the function to be projected with a step function defined on $n$ intervals and can be shown to have a computational complexity that is linear in $n$ (Best and Chakravarti, 1990).

Despite the numerical convenience of the aforementioned approximations, we believe that they are not suited for distributional data analysis. First and foremost, suppose that observations are given as probability density functions, so that one may want to interpret the results of a PCA, for instance, in terms of pdfs and not of quantile functions. If one were to estimate discontinuous principal directions through any of the algorithms in Ayer et al. (1955), it would not be possible to do so, as the corresponding cdfs would not be differentiable. In addition to that, the choice of the number of intervals $n$ is not obvious when quantile functions are not directly observed but obtained with transformation. If $n$ needs to be big to faithfully approximate the true quantile functions, this projection can be quite slow.

For these reasons, we propose to resort to a B-spline expansion, through which we can derive an alternative approximation of the projection operator $\Pi_X$, without incurring in the issues of the algorithms in Ayer et al. (1955). Moreover, we will also show in Section 6.5.3 that the proposed B-spline expansion also leads us to a simpler and faster reformulation of the geodesic PCA in Bigot et al. (2017).

### 6.4.3 Monotone B-splines representation

In what follows, let $\mu = U([0,1])$. Moreover, denote with $\boldsymbol{x} = [x_1, \dots, x_k]' \in \mathbb{R}^k$ a generic vector.

As already said, through the $\varphi_\mu$ map, we can identify $\mathcal{W}_2(\mathbb{R})$ with the space

$$L_2([0,1])^\uparrow := \{F^- \in L_2([0,1]) \text{ s.t. } F^- \text{ is monotonically nondecreasing}\}$$

This leads us to consider a suitable B-spline basis for the space, to efficiently evaluate all the computations needed in our algorithms and for a convenient way to express the constraints which define $L^2([0,1])^\uparrow$. In particular, we consider the basis of quadratic splines with equispaced knots in $[0,1]$. The reason for this particular choice is two-folded. First of all, splines of degree greater than one enjoy the nice property of uniform approximation of all continuous functions as the maximum distance between knots goes to zero, in turn this means that the closure of the linear space generated by the spline basis w.r.t the $L_2$ norm coincides with $L_2([0,1])$. Secondly, quadratic splines are particularly well suited to characterize monotonic functions by looking at the coefficients of the (quadratic) B-spline expansion, as shown in the next Proposition.

**Proposition 32.** *Let $\{\psi_j^k\}_{j=1}^J$ be a basis of B-splines of order $k$ defined over the knots $x_1, \dots, x_{J+k+2}$. Let $f(x) = \sum_{j=1}^J a_j \psi_j^k(x)$, then:*

1. *If the coefficients $\{a_j\}$ are monotonically increasing (decreasing) $f$ is monotonically increasing (decreasing)*
2. *If $k = 2$, then 1. holds with an "if and only if"*

Before proceeding, let us fix some notation. From now on, we omit the dimension index "$k$" for the spline basis, writing $\psi_j$ for $\psi_j^2$, moreover we will let $\{\psi_j\}_{j=1}^J$ with fixed $J > 0$ denote a B-spline basis in $L_2([0,1])$.

**Remark 27.** *Let $\mathbb{R}^{J\uparrow}$ be the set of vectors $v \in \mathbb{R}^J$ with nondecreasing coefficients. That is, letting $G = \{g_{ij}\}$ be the $J \times J$ binary matrix such that $\sum_j g_{ij} v_j = v_i - v_{i-1}$, for any element $\boldsymbol{v} \in \mathbb{R}^J$ it holds that $G\boldsymbol{v} \geqslant 0$. Using Proposition 32, through the coordinates operator, the set $L_2([0,1])^\uparrow \cap Span\{\psi_j\}_{j=1}^J$ is fully identifiable with $\mathbb{R}^{J\uparrow}$, endowed with the metric given by the symmetric positive definite matrix $E$ with entries*

$$E_{ij} = \langle \psi_i, \psi_j \rangle_{L_2([0,1])}. \tag{6.9}$$

*The norm induced is therefore $\|\boldsymbol{x}\|_E^2 = \boldsymbol{x}^T E \boldsymbol{x}$.*

**Remark 28.** *It is possible to find a basis for $\mathbb{R}^J$ with vectors lying in $\mathbb{R}^{J\uparrow}$ (and so in $X_J$), namely the vectors $(0,\dots,0,1)$, $(0,\dots,0,1,1)$ etc. In other words, $Span(L_2([0,1])^{\uparrow} \cap Span\{\psi_j\}_{j=1}^J) = Span\{\psi_j\}_{j=1}^J$ for every $J > 0$. This tells us that the convex cone of monotone splines is indeed quite big inside the spline space, and this a priori is beneficial for extrinsic methods, especially for PCA.*

From now on, to lighten the notation, we deliberately confuse the coefficients of the splines, living in $\mathbb{R}^J$ or $\mathbb{R}^{J\uparrow}$ (with the metric given by $E$), with the corresponding spline functions living in the subsets of $L_2([0,1])$ given by $L_2([0,1])^{\uparrow} \cap Span\{\psi_j\}_{j=1}^J$ and $Span\{\psi_j\}_{j=1}^J$.

**Remark 29.** *Lastly, we point out that $\mathbb{R}^{J\uparrow}$ has the structure of a convex polytope, since the constraints given by $G\boldsymbol{v} \geqslant 0$ (guaranteeing that $\boldsymbol{v} \in \mathbb{R}^{J\uparrow}$) are linear. Such geometric property makes optimization on $\mathbb{R}^{J\uparrow}$ handy and is key for the empirical methods developed in the remaining of the paper.*

As a consequence of Remark 29, the optimization problem given by the projection of a vector $\boldsymbol{v} \in \mathbb{R}^J$ onto $\mathbb{R}^{J\uparrow}$ can be formulated as follows:

$$\Pi_{\mathbb{R}^{J\uparrow}}(\boldsymbol{v}) = \underset{G\boldsymbol{w}\geqslant 0}{\arg\min} \|v - w\|_E. \tag{6.10}$$

The computational complexity required to solve (6.10) is at most cubic in the number of basis elements $J$ (Potra and Wright, 2000).

Preliminary analysis showed that solving the optimization problem in (6.10) compares favorably with the Pool Adjacent Violators Algorithm (PAVA) in Ayer et al. (1955). In particular, computing PAVA with $n = 100$ approximation intervals is roughly eight times slower than (6.10) with $J = 20$ (a reasonable choice, leading to negligible approximation error, in our examples, with a quadratic spline basis). Increasing $n = 1000$ for PAVA makes it 700 times slower than (6.10).

In addition to that, resorting to a discretized approximation of quantiles would also increase the cost of the projected PCA, due to the need of using some functional PCA implementation, as opposed to the low-dimensional multivariate model we are able to implement with the B-spline basis functions.

## 6.5 Empirical Models with B-splines

In this Section, we present the empirical counterparts of the projected PCA defined in Section 6.3 and provide an illustrative example of projected linear regression, namely when both the dependent and independent variables are distributions.

Let $\{\psi_j\}_{j=1}^J$ be a fixed quadratic B-spline basis. Upon approximating the observed quantile functions with their spline expansion, thanks to Remark 27, we can develop our methodology in $\mathbb{R}^J$, considering the metric induced by $E$ instead of the usual one.

Indeed, given a vector $\boldsymbol{w} \in \mathbb{R}^J$, we can identify the corresponding function in $L_2$ by the map $\boldsymbol{w} \mapsto \sum_{j=}^{J} w_j \psi_j$.

For the projected PCA in Section 6.5.1 and for the geodesic PCA in Section 6.5.3 we consider observations $F_1^-, \ldots, F_n^-$, and let $F_0^-$ be the centering point of the PCA. In our examples, $F_0^-$ will always be the barycenter of the observations. As a preprocessing step, we approximate each of these quantile functions through a B-spline expansion and denote by $\boldsymbol{a}_i = \{a_{ij}\}_j$ and $\boldsymbol{a}_0 = \{a_{0j}\}_j$ the coefficients of the spline representation associated to $F_i^-$ and $F_0^-$ respectively, that is, $F_i^- \approx \sum_{j=}^{J} a_{ij}\psi_j$. For the projected regression in Section 6.5.2, let observations $\{(F_z^-, F_y^-)_i\}_{i=1}^n$, where the $F_{zi}^-$'s are realizations of the independent variable $\mathcal{Z}$ and the $F_{yi}^-$'s are realizations of the dependent variable $\mathcal{Y}$. We apply the same preprocessing step and let $\boldsymbol{a}_i^{(z)}$ and $\boldsymbol{a}_i^{(y)}$ denote the coefficient of the spline approximation of $F_{zi}^-$ and $F_{yi}^-$ respectively.

### 6.5.1 Empirical PCA

Denote with $A$ the $(n \times J)$ matrix with rows $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$. As in standard PCA, the first principal component centered in $\boldsymbol{a}_0$ is found by solving the optimization problem:

$$\boldsymbol{w}_1^* = \underset{\boldsymbol{w}:\|w\|_E=1}{\arg\max} \sum_i |\langle \boldsymbol{a}_i - \boldsymbol{a}_0, \boldsymbol{w} \rangle_E|^2 = \underset{\boldsymbol{w}:\|\boldsymbol{w}\|_E=1}{\arg\max} \|AE\boldsymbol{w}\|^2 \tag{6.11}$$

where $A$ is the matrix whose i–th row is given $\boldsymbol{a}_i - \boldsymbol{a}_0$. The optimization problem (6.11) can be solved similarly to a Rayleigh quotient: using Lagrange multipliers, (6.11) is equivalent to

$$\mathcal{L}(\boldsymbol{w}) := \boldsymbol{w}^T (A\ E)^T\ A\ E\ \boldsymbol{w} - \lambda(\boldsymbol{w}^T\ E\ \boldsymbol{w} - 1) \tag{6.12}$$

Deriving (6.12) w.r.t $\boldsymbol{w}$ and equating the derivative to zero shows that the solutions to $d\mathcal{L}(\boldsymbol{w})/d\boldsymbol{w} = 0$ are the eigenvectors of the matrix $A^T AE$. Hence, ordering the eigenvalues of $A^T AE$ in decreasing order, the first principal component $\boldsymbol{w}_1^*$ corresponds to the first eigenvector. Using similar arguments it can be shown that $\boldsymbol{w}_2^*, \ldots \boldsymbol{w}_J^*$ correspond to the remaining eigenvectors.

Once the first $k$ principal directions $\boldsymbol{w}_1^*, \ldots, \boldsymbol{w}_k^*$ are found, the projection of a new observation $x^* = \sum_{j=1}^{J} a_j^* \psi_j$ onto $U_X^{k,x_0}$ (see Definition 35) is found exploiting Proposition 30. In particular, the following optimization problem is to be solved:

$$\begin{aligned} &\underset{\lambda_j \in \mathbb{R}}{\arg\min} \|(\langle \boldsymbol{a}^* - \boldsymbol{a}_0, \boldsymbol{w}_i^* \rangle_E - \lambda_i)_{i=1}^k\| \\ &\text{s.t. } G\Big(\sum_{i=1}^{k} \lambda_i \boldsymbol{w}_i^* + \boldsymbol{a}_0\Big) \geqslant 0 \end{aligned} \tag{6.13}$$

which is equivalent to the minimization of a norm inside a polytope, that is a well-studied problem in $\mathbb{R}^J$ (see Sekitani and Yamamoto, 1993) and there exist a variety of fast numerical routines to solve it.

### 6.5.2 Empirical Regression

In this section, we provide the details of the estimation procedure for a projected regression model where both the independent and the dependent variables are distribution-valued. It is straightforward to extend our methodology to cases when only one of these variables is distribution-valued and the other one takes values in $\mathbb{R}^q$.

First, we outline how to obtain an estimator for the linear operator $\Gamma$ in Definition 38. Following Section 6.3.2 we first embed both $\mathcal{Y}$ and $\mathcal{Z}$ in $L_2([0,1])$ through the inclusion operator $L_2([0,1])^{\uparrow} \hookrightarrow L_2([0,1])$, and assume the functional linear model presented in Ramsay (2004) and Prchal and Sarda (2007)

$$\mathcal{Y}(t) = \alpha(t) + \int_0^1 \beta(t,s)\mathcal{Z}(s)ds + \varepsilon(t), \qquad t \in [0,1] \tag{6.14}$$

so that $\Gamma = \Gamma_{\alpha,\beta}$ is the operator $\Gamma_{\alpha,\beta}(v)(t) = \alpha(t) + \int_0^1 \beta(t,s)v(s)ds$. The goal is then to estimate $\alpha \in L_2([0,1])$ and $\beta \in L_2([0,1]^2)$. Further, we assume that $\varepsilon$ and $\mathcal{Z}$ are uncorrelated: $\mathbb{E}[\mathcal{Z}(s)\varepsilon(t)] = 0$ for every $t, s \in [0,1]$.

Consider now observations $\{(F_z^-, F_y^-)_i\}_{i=1}^n$ and the corresponding spline coefficients. Further, we project $\alpha(t)$ on the same spline basis, so that $\alpha \approx \sum_{j=1}^J \theta_{\alpha j}\psi(j)$ and $\beta(t,s)$ on the basis on $[0,1]^2$ with $J \times J$ elements, so that $\beta(t,s) \approx \sum_{i,j\prime=1}^J \Theta_{\beta ij}\psi_i(t)\psi_j(s)$. Neglecting the spline approximation error, model (6.14) entails

$$\boldsymbol{a}_i^{(y)} = \boldsymbol{\theta}_\alpha + \Theta_\beta E\boldsymbol{a}_i^{(z)} + \boldsymbol{a}_i^{(\varepsilon)}, \qquad i = 1, \dots, n \tag{6.15}$$

where $\boldsymbol{a}_i^{(\varepsilon)}$ denotes the spline expansion coefficients of the unobserved error $\varepsilon_i(t)$.

We propose to estimate (6.15) using the same approach of Prchal and Sarda (2007), but extending it to account for spline approximations for both dependent and independent variables. We focus only on the estimate $\widehat{\Theta}_\beta$ of $\Theta_\beta$ since once such estimate is obtained, the estimate for $\boldsymbol{a}_\alpha$ can be straightforwardly derived, (see Cai and Hall, 2006) as:

$$\hat{\boldsymbol{\theta}}_\alpha = \overline{\boldsymbol{a}^{(y)}} - \widehat{\Theta}_\beta E\overline{\boldsymbol{a}^{(z)}}$$

where $\overline{\boldsymbol{a}^{(y)}}$ and $\overline{\boldsymbol{a}^{(z)}}$ are the means of $\boldsymbol{a}^{(y)}$ and $\boldsymbol{a}^{(z)}$ respectively.

The estimator $\widehat{\Theta}_\beta$ is found by penalized least square minimization:

$$\widehat{\Theta}_\beta = \arg\min_{\Theta} \frac{1}{n}\sum_{i=1}^n \| \left(\boldsymbol{a}_i^{(y)} - \overline{\boldsymbol{a}^{(y)}}\right) - \Theta E\left(\boldsymbol{a}_i^{(z)} - \overline{\boldsymbol{a}^{(z)}}\right) \|^2 + \rho\text{Pen}(1,\Theta) \tag{6.16}$$

where $\rho > 0$ is a penalization parameter to be fixed (usually through cross-validation) and $\text{Pen}(1,\Theta)$ is a penalization term defined in Prchal and Sarda (2007).

Briefly, the term $\text{Pen}(1,\Theta)$ in (6.16) penalizes both the norm of $\beta(t,s)$ and its derivatives, thus favoring smoother solutions. As shown in Prchal and Sarda (2007), (6.16) has a closed form solution. Nonetheless, the form of our solution differs from the one

presented in Prchal and Sarda (2007), since they work directly on discretized functions while we propose to estimate spline coefficients, and some care must be taken since they can use (up to scaling) the usual inner product in the Euclidean space of discretized functions, while we must consider the inner product induced by $E$. However, the procedure for obtaining our result is identical to the one in Prchal and Sarda (2007). Hence, we only report the expression for the estimate.

Let $\hat{C}$ be the matrix with entries

$$\hat{C}_{ks} = \langle \frac{1}{n} \sum_{i=1}^{n} \langle \boldsymbol{a}_i^{(z)}, b_k \rangle_E \, \boldsymbol{a}_i^{(z)}, b_s \rangle_E,$$

where $b_k$ and $b_s$ are the $k$-th and $s$-th elements of the standard Euclidean basis in $\mathbb{R}^J$. Further let $\hat{D}$ the matrix with entries

$$\hat{D}_{ks} = \langle \frac{1}{n} \sum_{i=1}^{n} \langle \boldsymbol{a}_i^{(z)}, b_k \rangle_E \, \boldsymbol{a}_i^{(y)}, b_s \rangle_E.$$

Finally, let $E'$ denote the matrix with entries $E'_{ij} =< \psi'_i, \psi'_j >$ (where $\psi'_i$ denotes the first derivative of the B-spline basis function $\psi_i$), $C_\rho = E^T \otimes (\hat{C} + \rho E')$, and $P = E'^T \otimes E + E^T \otimes E'$, where $\otimes$ denotes the Kronecker product. Then the solution of (6.16) can be expressed as

$$\text{vec}(\widehat{\Theta}_\beta) = (C_\rho + \rho P)^{-1} \text{vec}(\hat{D})$$

where $\text{vec}(\cdot)$ denotes the *vectorization* of the matrix.

Finally, our projected regression model is the composition of the operator induced by $(\hat{\boldsymbol{\theta}}_\alpha, \hat{\Theta}_\beta)$ with the projection on $\mathbb{R}^{\uparrow J}$:

$$\mathbb{E}[\boldsymbol{a}_i^{(y)} \mid \boldsymbol{a}_i^{(z)}] = \Gamma_{\text{P}}(\boldsymbol{a}_i^{(z)}) = \Pi_{\mathbb{R}^{J\uparrow}} \left( \hat{\boldsymbol{\theta}}_\alpha + \widehat{\Theta}_\beta E \boldsymbol{a}_i^{(z)} \right).$$

### 6.5.3 An alternative optimization routine for the geodesic PCA and a comment on the computational costs

We now show how the framework in Section 6.4 can be employed also to derive faster numerical algorithms to find the global and nested geodesic PCA as of Definition 39 and Definition 40.

**Proposition 33.** *(Global geodesic PCA) A $k$ dimensional global geodesic PC centered in $\boldsymbol{a}_0$ is the subset of $\mathbb{R}^{J\uparrow}$ spanned by $\{\boldsymbol{w}_1, \cdots, \boldsymbol{w}_k\}$, linearly independent, which solve:*

$$\begin{aligned} \underset{\{\boldsymbol{\lambda}_i\}_1^n, \{\boldsymbol{w}_j\}_1^k}{\arg\min} & \sum_{i=1}^{n} ||\boldsymbol{a_i} - \boldsymbol{a_0} - \sum_{j=1}^{k} \lambda_{ij} \cdot \boldsymbol{w}_j||_E^2 \\ & \textit{s.t. } G\Big( \sum_j \lambda_{ij} \boldsymbol{w}_j + \boldsymbol{a}_0 \Big) \geqslant 0 \end{aligned} \tag{6.17}$$

**Proposition 34.** *(Nested geodesic PCA) With the same notation as above, a $k$ dimensional nested geodesic PC, centered in $\boldsymbol{a}_0$ is the set spanned by $\{\boldsymbol{w}_1, \cdots, \boldsymbol{w}_k\}$ in $\mathbb{R}^{J\uparrow}$, where the $\boldsymbol{w}_i$s are found recursively from $\boldsymbol{w}_1$ to $\boldsymbol{w}_k$, such that $\boldsymbol{w}_h$ is a solution, for every $h$, of:*

$$\begin{aligned} &\underset{\{\boldsymbol{\lambda}_i\}_{i=1}^n, \boldsymbol{w}}{\arg\min} \sum_{i=1}^{n} \|\boldsymbol{a_i} - \boldsymbol{a_0} - \lambda_i \boldsymbol{w}\|_E^2 \\ &\qquad \textit{s.t. } \langle \boldsymbol{w}_j, \boldsymbol{w} \rangle_E = 0, \quad j = 1, \ldots, h-1 \\ &\qquad\qquad G\Big(\lambda_i \boldsymbol{w} + \boldsymbol{a_0}\Big) \geqslant 0, \quad \|\boldsymbol{w}\|_E = 1 \end{aligned} \tag{6.18}$$

To solve (6.17) and (6.18) we employ an interior point method using the solver Ipopt (Waechter and Biegler, 2006). When comparing our implementation with $J = 20$ spline basis and the one in Cazelles et al. (2018), we notice a substantial performance improvement, by a factor of 35 for a data set of $n = 100$ distributions, due to the fact working with spline approximations reduces greatly the number of parameters in the optimization problem.

Further, note that (6.17) and (6.8) seem extremely similar. However, in (6.8) the optimization is carried out having fixed $\boldsymbol{w}_1^*, \ldots, \boldsymbol{w}_k^*$ and for a single observation, while in (6.17) the optimization is done over a much larger set of parameters. In fact, the number of parameters in (6.17) is $(n+k)J$, hence the computational complexity needed to solve (6.17) is cubic in both the number of bases and the number of observations. On the other hand, the projected PCA requires a linear time in the number of observations (computation of $A^T A E$) and cubic time in the number of basis $J$ (eigendecomposition and projections of new observations).

## 6.6 Asymptotic Properties

In this section, we study the convergence of the proposed projected empirical methods. First of all, we show that as the number of spline basis $J$ increases, the error due to the spline approximation vanishes if the data is sufficiently regular. Further, under a suitable set of assumptions, we establish consistency results for the projected PCA and for the projected distribution on distribution regression.

### 6.6.1 Convergence of Quadratic B-splines

In the following, denote with $W_k^r([0,1])$ the space of functions whose weak derivatives up to order $k$ belong to $L_r([0,1])$, further denote with $D$ the (weak) derivative operator, so that $Df = f'$, $D^2 f = f''$ and so on,

**Proposition 35.** *Let $\mu$ a probability measure on $\mathbb{R}$, $F_\mu^-$ its quantile function such that $F_\mu^- \in W_3^\infty$. For each $J$ let $\{\psi_j\}_{j=1}^J$ denote a quadratic B-spline basis on $J$ equispaced*

*knots in* $[0,1]$. *Then there exist a sequence of spline functions* $S_J = \sum_{j=1}^{J} \lambda^{(J)} \psi_j^{(J)}$, *with* $\lambda_j^{(J)}$ *monotonically non-decreasing in* $j$ *for every* $J$, *such that:*

$$\|S_J - F_\mu^-\|_\infty \leqslant C \|D^2 f_\mu^-\|_\infty J^{-2}$$

*with* $f_\mu^- = DF_\mu^-$ *and* $C > 0$ *constant.*

Let us remark two important facts.

**Remark 30.** *Since the inclusion* $L_\infty([0,1]) \subset L_2([0,1])$ *is continuous, thanks to Hölder inequality, the convergence rates hold also for the* $L_2$ *norm. By default we will use the* $L_2$ *norm if not stated differently.*

**Remark 31.** *By Poincaré inequality, if* $\|D^3 f\|_\infty < C$ *then* $f$ *belongs to a sphere in* $W_3^\infty([0,1])$ *whose radius depends on* $C$ *and on the Poincaré constant of* $[0,1]$*; viceversa, all the elements in the sphere of radius* $C$ *in* $W_3^\infty([0,1])$ *clearly have (weak) derivatives bounded by* $C$.

### 6.6.2 Consistency

In this Section we prove the consistency of the projected methods under some assumptions on the data-generating process. In particular, we show that that there exists a number of basis functions $J > 0$ and a sample size $n$ such that the error committed by the empirical models in Section 6.5 is smaller than $\varepsilon > 0$, for any fixed $\varepsilon$.

#### PCA

Consistency of spline-based PCA for functional data has been addressed, among the first, by Silverman et al. (1996) and Qi and Zhao (2011). As one of the main building blocks of our projected PCA is the PCA in the ambient space, that is $L_2([0,1])$, it is natural to follow Qi and Zhao (2011) in making the following assumptions. Consider data $\mu_1, \ldots, \mu_n$, $F_1^-, \ldots, F_n^-$ the corresponding quantile functions, then:

(P1) The data generating process satisfies $F_1^-, \ldots, F_n^- \sim \mathcal{F}$ with the $F_i^-$ independent and $\mathbb{E}[\mathcal{F}] = 0$.

(P2) $F_1^-, \ldots, F_n^-$ can be approximated by functions in $W_3^\infty$ with uniformly bounded third derivative.

(P3) $\mathbb{E}[\|F_i^-(t)\|^4] < \infty,\ i = 1, \ldots, n$.

(P4) The eigenvalues of the covariance operator of $\mathcal{F}$ have multiplicity 1.

(P5) The eigenfunctions of the covariance operator of $\mathcal{F}$ belong to some bounded set in $W_3^\infty([0,1]) \subset W_3^2([0,1])$.

Before stating the main results, let us comment on assumptions (P1)-(P5). First of all, (P2) is essential in order to apply Proposition 35 and get uniform errors on the data set. Moreover, (P2) is satisfied, for instance, if the $F_i^-$'s lie in the $L_2$-closure of a ball of radius $M > 0$ in $W_3^\infty$. (P4) is a rather standard condition and is satisfied if $\mu_1, \dots, \mu_n \in \mathcal{W}_4(\mathbb{R})$. (P4) and (P5) imply the assumptions that in Qi and Zhao (2011) are used for the consistency results. In particular, (P5) is stronger than the corresponding assumption in Qi and Zhao (2011), where the eigenfunctions are assumed to belong to $W_2^2([a,b])$. Similarly, in such work, there is no counterpart of assumption (P2); in fact we need these stronger regularity conditions to get uniform errors when using B-splines. Still some of the examples Qi and Zhao (2011) provide of situations satisfying their assumptions, meet also our requirements. Finally, the zero-mean assumption in (P1) might seem a little odd, since we know that the quantile functions are monotonically nondecreasing. However, observe that it is always possible to subtract the empirical mean from the observations to satisfy (asymptotically) this assumption.

Let $J$ denote the dimension of a quadratic B-spline basis on $[0,1]$ and let $\boldsymbol{a}_i^J$ the coefficients of the B-spline approximation of $F_i^-$. In what follows, to lighten the notation, we refer to a set of spline coefficients both as elements of $\mathbb{R}^J$ with the $E$-norm, or as functions in $L_2$, without making explicit reference to the coordinate operator and its inverse.

**Proposition 36.** *Under assumptions (P1)-(P5), for any $\varepsilon > 0$ there exists a sample size $n > 0$ and a number of basis functions $J > 0$ such that:*

$$\left| \max_{\|w\|_{L_2}=1} \frac{1}{n} \sum_i \langle F_i^-, w \rangle_{L_2}^2 - \max_{\|\boldsymbol{w}\|_E=1} \frac{1}{n} \sum_i \langle \boldsymbol{a}_i^J, \boldsymbol{w} \rangle_E^2 \right| < K\varepsilon$$

*for some constant $K > 0$.*

Proposition 36 ensures the consistency of the B-spline approximation of the PCA for monotone functional data in $H$ which is equivalent to the consistent estimation of the projected principal directions.

Suppose now to have computed $U_k^J = \{\boldsymbol{w}_h^{J*}\}_{h=1}^k$, that is the approximations of the principal directions $U_k = \{\boldsymbol{w}_h^*\}_{h=1}^k$ found with $J$ basis functions. We observe that $Sp(U_k^J) \cap L_2([0,1])^\uparrow = Sp(U_k^J) \cap \mathbb{R}^{J\uparrow}$. Since for any set of coefficients $\lambda_h$ we have the convergence $\sum \lambda_h \boldsymbol{w}_h^{J*} \to \sum \lambda_h w_h^*$, we obtain that the projection of a point onto $Sp(U_k^J) \cap L_2([0,1])^\uparrow$ converges to the projection onto $Sp(U_k) \cap L_2([0,1])^\uparrow$. Thus we also have convergence of the projection onto the principal components.

### Regression

We consider model (6.14) given samples $\{(F_z^-, F_y^-)_i\}_{i=1}^n$. We make the following assumptions:

(R1) The data generating process satisfies (6.14) and $\mathbb{E}[\mathcal{Z}(s)\varepsilon(t)] = 0$ for every $t, s \in [0,1]$.

(R2) $\alpha \in L_2([0,1])$ and $\beta \in L_2([0,1] \times [0,1])$.

(R3) With probability 1, each quantile function in the samples $\{(F_z^-, F_y^-)_i\}_{i=1}^n$ lies inside a sphere of radius $K > 0$ in $W_\infty^3([0,1])$.

Without loss of generality, suppose that both the dependent and the independent variables have been centered by subtracting their mean so that $\mathbb{E}[\mathcal{Z}] = \mathbb{E}[\mathcal{Y}] = 0$ and $\alpha = 0$.

The strategy to prove the consistency of the projected linear regression is the following. First of all, we prove that the estimator $\widehat{\Theta}_J$ converges to the estimator $\widehat{\Theta}_{\text{PS}}$, defined in Prchal and Sarda (2007), for large enough $n$ and $J$. Second, we exploit the consistency of the estimator in Prchal and Sarda (2007) combined with the approximation results of the metric projection, to establish consistency in terms of the prediction error of our projected regression operator.

Briefly $\widehat{\Theta}_{\text{PS}}$ is obtained by minimizing an objective function similar to the one in (6.16), but where the spline approximation is used only for $\Theta$, while the $F_{zi}^-$'s and the $F_{yi}^-$'s are assumed fully observed, and not approximated through splines. Calling $B$ the vector of functions with entries $\psi_1, \ldots, \psi_J$, $\widehat{\Theta}_{\text{PS}}$ is defined as:

$$\widehat{\Theta}_{\text{PS}} = \arg\min_{\Theta} \frac{1}{n} \sum_i \|F_{yi}^- - \langle F_{zi}^-, B^T \Theta B \rangle\|^2 + \rho \text{Pen}(1, \Theta).$$

Convergence of $\widehat{\Theta}_J$ to $\widehat{\Theta}_{\text{PS}}$ is shown in the next proposition

**Proposition 37.** *Under assumptions (R1)-(R3), if the number of samples is big enough $\widehat{\Theta}$ and $\widehat{\Theta}_J$ exists with probability close to 1, and there is $J > 0$ such that $\|\widehat{\Theta}_{PS} - \widehat{\Theta}_J\|_{E \otimes E} < \varepsilon$.*

Let $\widehat{\beta}_{\text{PS}}$ and $\widehat{\beta}_J$ be the kernels $\widehat{\beta}_{\text{PS}} = B^T \widehat{\Theta}_{\text{PS}} B$ and $\widehat{\beta}_J = B^T \widehat{\Theta}_J B$. Since $\|\widehat{\beta}_{\text{PS}}(s,t) - \widehat{\beta}_J(s,t)\|_{L_2([0,1]^2)} = \|\widehat{\Theta}_{\text{PS}} - \widehat{\Theta}_J\|_{E \otimes E}$, we established strong convergence of our kernel to the estimator of Prchal and Sarda (2007). This implies that the consistency results for the estimator $\widehat{\Theta}_{\text{PS}}$ holds also for $\widehat{\Theta}_J$, with respect to the seminorm induced by the covariance operator of $\mathcal{Z}$.

Specifically, given $\mathcal{Z}$ $H$-valued random variable and its covariance operator $\mathcal{C}_{\mathcal{Z}}$, for any $\varphi \in L_2([0,1]^2)$, we consider the semi-norm on $L_2([0,1]^2)$ given by:

$$\|\varphi\|_{\Gamma_{\mathcal{Z}}} = \int_{[0,1]} \langle \mathcal{C}_{\mathcal{Z}} \varphi(\cdot, t), \varphi(\cdot, t) \rangle dt$$

Thus, the following result is immediately implied since strong convergence implies seminorm convergence (see Appendix 6.11).

**Corollary 7.** *For $J > 0$ big enough $\mathbb{E}[\|\beta - \widehat{\beta}_J\|_{\mathcal{C}_{\mathcal{Z}}}] < \varepsilon$.*

*Proof.* We use the seminorm triangle inequality:

$$\|\beta - \widehat{\beta}_J\|_{\mathcal{C}_{\mathcal{Z}}} \leqslant \|\beta - \widehat{\beta}\|_{\mathcal{C}_{\mathcal{Z}}} + \|\widehat{\beta} - \widehat{\beta}_J\|_{\mathcal{C}_{\mathcal{Z}}}.$$

The first term on the right hand side converges to zero thanks to Theorem 2 in Prchal and Sarda (2007), while the second term converges to zero thanks to Proposition 37 and the previous observations. □

Lastly, we need to take into account the projection step. First, we notice that $\|\beta - \widehat{\beta}\|_{\Gamma_Z}$ corresponds to the expected prediction error, in fact, as in Prchal and Sarda (2007):

$$\|\beta - \widehat{\beta}_J\|_{\mathcal{C}_{\mathcal{Z}}} = \int_{[0,1]} \mathbb{E}\left[\langle \mathcal{Z}, \beta(\cdot,t) - \widehat{\beta}_J(\cdot,t)\rangle^2 \mid \widehat{\beta}_J\right] dt,$$

further, by Hölder's inequality $\mathbb{E}\left[|\langle \mathcal{Z}, \beta - \widehat{\beta}_J\rangle| \,\middle|\, \widehat{\beta}_J\right] \to 0$, which straightforwardly yields $\mathbb{E}\left[\|\Gamma_\beta(z) - \Gamma_{\widehat{\beta}_J}(z)\| \,\middle|\, \widehat{\beta}_J\right] \to 0$.

Thus, the following simple lemma ensures the consistency of the spline approximation of the projection on $X$ and leads to the consistency of the projected regression in terms of prediction error. Again, following Remark 27, we can identify the space monotone $B$-splines with $J$ basis functions with $\mathbb{R}^{J\uparrow}$. Hence, to lighten the notation, we denote $\Pi_{\mathbb{R}^{J\uparrow}}$ the metric projection operator onto the space of monotone $B$-splines with $J$ basis functions.

**Lemma 13.** *Given $\beta_n \to \beta$ in $H$, for any $\varepsilon > 0$ there exists $n, J > 0$ such that $\|\Pi_{\mathbb{R}^{J\uparrow}}(\beta_n) - \Pi_{L_2([0,1])^\uparrow}(\beta)\| \leqslant \varepsilon$.*

## 6.7 Numerical Illustrations for the PCA

In this section we perform PCA on different simulated data sets and on a real data set of Covid-19 mortality data in the US. In particular, on the simulated data sets we compare the performance of our projected PCA (in terms of approximation error and interpretability of the directions) with the ones of intrinsic methods, showing that the projected PCA is a valid competitor in a diverse set of situations. For the Covid-19 data set, we compare inference obtained using the projected, nested and log PCA, highlighting the practical benefits of the projected PCA over the log one.

For the projected, nested and global PCAs we need to fix a B-spline basis to express the quantile functions. In particular, we fix an equispaced quadratic B-spline basis with $J$ interior knots on $[0,1]$. Here, the number of basis $J$ is always fixed to 20, which provided a negligible approximation error of the quantile functions. We did not observe any appreciable change when increasing it. In Appendix 6.13 we show further simulations where we perform sensitivity analysis as the number of basis increases for a fixed sample size, we provide empirical confirmation of the consistency results in Section 6.6 and give practical guidance on how to choose $J$.

### 6.7.1 Simulation studies

We consider three different simulations to compare both the interpretability and the ability to compress information of different PCAs.

We compare our projected PCA with the nested and global geodesic PCAs (Bigot et al., 2017; Cazelles et al., 2018) and the *simplicial* PCA (Hron et al., 2014).

Briefly, the simplicial PCA applies a transformation that maps densities defined on the same compact interval $I$ into functions in $L_2(I)$, called *centered log ratio*. Then, a standard $L_2$ PCA is performed on the transformed pdfs and, by the inverse of the centered log ratio transform, the results are mapped back to the space of densities, called Bayes space (for a more accurate definition, see Egozcue et al., 2006). In particular, we remark that, to be well defined, the simplicial PCA requires that all the pdfs have support equal to $I$, which is a strong assumption in practice. Further details about simplicial PCA are given in Appendix 6.12.

As for the projected PCA, to compute the simplicial PCA, we resort to a B-spline approximation, but this time of the transformed pdfs. Hence, we need to select a B-spline basis on the support of the pdfs $I$. In this case, we fix a cubic B-spline basis with

$$J' = J = 20$$

interior knots on $I$, as this choice yielded a negligible approximation error for the transformed pdfs.

In the first scenario, we simulate data from

$$\begin{aligned} p_i(x) &\propto \frac{1}{\sigma_i} \exp\left((x-\mu_i)^2/(2\sigma_i^2)\right) \mathbb{I}(x \in [-10, 10]), \quad i = 1, \dots 100 \\ \mu_i &\sim 0.5\mathcal{N}(-3, (0.2)^2) + 0.5\mathcal{N}(3, (0.2)^2) \\ \sigma_i &\sim \text{Uniform}([0.5, 2.0]) \end{aligned} \tag{6.19}$$

Where "proportional to" stands for the fact that we confine the density to the support $[-10, 10]$ and renormalize it so that it integrates to $1$.

Observe that there are two sources of variability across the pdfs from the data generating process (6.19). The first one is the location of the *peak* $\mu_i$ and the second one is the *width* of the distribution around the peak, controlled by $\sigma_i$. See Figure 6.3.

Figure 6.4 shows the first two principal directions obtained using the different methods. We can notice several differences between them. Focusing on the first principal direction, we can see that the simplicial, projected and nested PCAs detect a change in the location of the peak of the pdf. In particular, the first direction for the Wasserstein PCAs represents a shift from left to right of this peak, while for the simplicial PCA the first direction is associated to a peak in $3$ (blue lines, negative values of the scores) or to a peak in $-3$ (red lines, positive value of the scores). This also highlights the difference in the geometries underlying the Wasserstein and Bayes spaces. Looking at the second principal direction instead, we can see how in the Wasserstein PCAs it clearly represents

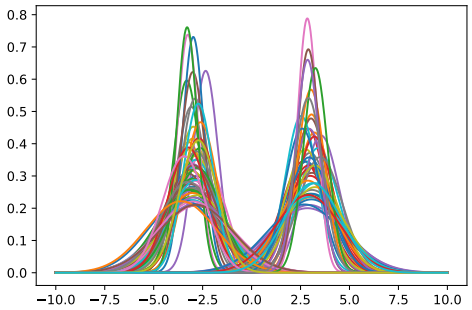



**Figure 6.3:** *Data set of pdfs generated from* (6.19)

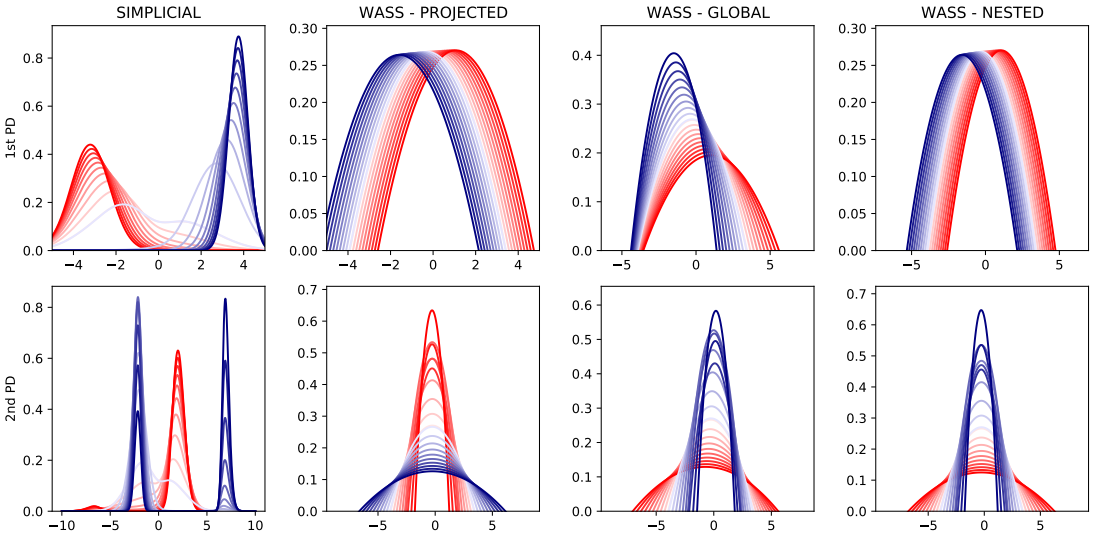



**Figure 6.4:** *Top row: first principal direction. Bottom row: second principal direction. Each line represents the pdf associated to* $\lambda \mathbf{w}_i$ *where* $\mathbf{w}_i$ *is the* $i$*–th principal direction* ($i = 1, 2$) *and* $\lambda$ *is a score ranging from* $-2$ *(darkest blue) to* $+2$ *(darkest red).*

a change in the width of the distribution, while for the simplicial PCA the interpretation is somewhat obscure.

The global geodesic PCA deserves a separate discussion. Indeed, from Definition 39 it is clear that a global principal component is a convex set without any notion of preferential directions, so that it is not possible to interpret separately the variation along the first and second direction found by the global PCA.

Now we present two additional simulations that quantify the amount of information that is "lost" by performing the PCA. As a metric, we consider the reconstruction error, that is, the quantity

$$RE_k = \frac{1}{n}\sum_{i=1}^{n} W_2(F_i^-, \widetilde{F}_i^-) \tag{6.20}$$

where the $F_i^-$'s are the observed probability measures, $\widetilde{F}_i^-$ are the reconstructed ones and $k$ is the dimension of the principal component. More in detail $\widetilde{F}_i^-$ is found by first projecting $(F_i^- - F_0^-)$ into $\mathbb{R}^k$ using the PCA and then applying the inverse transformation. Informally, the reconstruction error is a measure of the quantity of information lost by applying the PCA as a black-box dimensionality reduction.

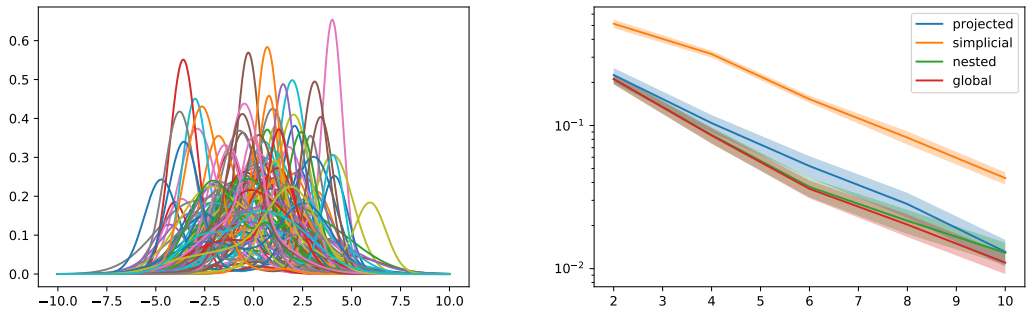
As evident in Equation (6.20), we measure the performance of PCAs just in terms of Wasserstein metric. This is likely to favor the performance of the Wasserstein PCAs over the simplicial one. Thus, the interesting performance comparison is the one between the geodesic PCAs and the projected PCA. Nevertheless, we think that is worth reporting also the results for the simplicial PCA, which is an intrinsic method in the Bayes space, to show that the underlying metric structures are extremely different. This also helps to appreciate the results in Section 6.8. Given the difference in the metric structure between Wasserstein and Bayes spaces, we believe that the choice between simplicial and Wasserstein frameworks is not trivial and should be application-driven.

To measure raw performance differences between geodesic and projected PCAs, we simulate data so that there is little recognizable structure in them, unlike in the previous example. The data generating process is as follows:

$$\begin{aligned} p_i(x) &\propto \sum_{j=1}^{K} w_{ij} \frac{1}{\sigma_{ij}} \exp\left((x-\mu_{ij})^2/(2\sigma_{ij}^2)\right) \mathbb{I}(x \in [-10,10]) + 10^{-5}, \quad i = 1, \dots 100 \\ \boldsymbol{w}_i &\sim \text{Dirichlet}_K(1/K) \\ (\mu_{ij}, \sigma_{ij}) &\sim \mathcal{N}(d\mu_{ij}; 0, 2^2)\text{Uniform}(d\sigma_{ij}, 0.5, 2.0) \end{aligned} \tag{6.21}$$

Observe that (6.21) is a finite dimensional approximation of the Dirichlet Process mixture model, a popular workhorse in Bayesian nonparametric statistics, that is well known to be dense in the space of densities on $\mathbb{R}$, see for instance Ferguson (1983). An example of the kind of pdfs generated from (6.21) is shown in Figure 6.5(a).

To separate the effect of the B-spline smoothing procedure, in this scenario we evaluate the reconstruction error in (6.20) considering $\widetilde{\mu}_i$ to be the reconstructed quantile




**(a)** **(b)**

**Figure 6.5:** *Left panel: example of simulated data set for Scenario 2. Right panel: reconstruction error as a function of the dimension of the principal component employed for the different methods. The solid lines represent the mean of 10 independent runs on independent data sets from* (6.21) *and the shaded area represent* $\pm$ *one standard deviation.*

functions (for the Wasserstein PCAs) or pdfs (for the simplicial PCA) and $\mu_i$ to be the probability measure represented by the B-spline approximation of the quantile function or the (centered log ratio of) the pdf respectively.

Figure 6.5(b) shows the reconstruction error as a function of the dimension of the principal component, that is, $RE_k$ as a function of $k$. We can see how the three Wasserstein PCAs consistently outperform the simplicial one. Moreover, as to be expected, the global geodesic PCA obtains the lowest reconstruction error for all the choices of dimension $k$, with the nested geodesic PCA being a close runner-up. However, the computational cost of finding the nested or global geodesic PCA can become prohibitive as the sample size or the number of bases in the B-spline expansion or the dimension $k$ increases. For comparison, finding the 10-dimensional projected PCA is around 1,000 times quicker than finding the corresponding global geodesic PCA and 200 times quicker than finding the nested geodesic one.

As an additional simulation, in Appendix 6.13 we investigate the effect of the number of B-spline basis $J$. In particular, we conclude that, for a fixed dimension $k$ the reconstruction error (6.20) increases with the number of basis functions, both for the projected and the simplicial PCA. Furthermore, we also observe that the reconstruction error for the simplicial PCA exhibits a larger variance than the reconstruction error for the projected PCA. Our insight is that this is due to the different degree of smoothness of the pdfs and the quantile functions. Since the quantile functions are in general smoother than the pdfs, their B-spline expansion should have lower variance.

### 6.7.2 Assessing the reliability of the projected PCA

A classical measure of performance of the standard Euclidean PCA, also useful to determine the dimension of the principal component to use, is the proportion of the explained variance. For a $k$-dimensional Euclidean principal component, this quantity is easily computed as a ratio of eigenvalues: $\sum_{j=1}^{k} \lambda_j / \sum_{j \geqslant 1} \lambda_j$. Upon truncating the series at the denominator, the same quantity can also be computed for PCA in infinite dimensional Hilbert spaces.

Due to the projection step involved in our definition of PCA, we argue that the proportion of explained variance might not be a reliable indicator of performance, nor should it be used to guide the choice of the dimension $k$. Instead, we propose a fast alternative based on the Wasserstein distance that we believe better represents the properties of the projected PCA, that is, the normalized reconstruction error:

$$NRE_k = \frac{\frac{1}{n}\sum_{i=1}^{n} W_2(F_i^-, \widetilde{F}_i^-)}{\frac{1}{n}\sum_{i=1}^{n} W_2(F_i^-, F_0^-)},$$

where the numerator corresponds to the reconstruction error in (6.20) and the denominator is the average distance between the observed measures and their barycenter. Observe that in Euclidean spaces, this quantity is closely related to the proportion of explained variance, since in Euclidean spaces maximizing variance in a subspace, amounts to minimizing the average distance from the subspace to data points.

Given its extrinsic nature, for a fixed dimension, the projected PCA might sometimes fail to capture the variability of some particular data set and, in those situations, an intrinsic approach should be preferred. However, given the high computational cost associated to geodesic PCAs, one would carry out such analysis only knowing that the results would be significantly better than the ones obtained by projected PCA. This calls for discerning whether the poor performance of projected PCA is due to its extrinsic nature or rather to the scarceness of structure in the data set under consideration: in the former situation it is likely that a geodesic approach would yield better results, in the latter instead, it is likely that results remain the same.

We propose now two empirical indicators of the "reliability" of the empirical projected PCA. The first one measures, once a $k$-dimensional principal component is found, how reliable are the projected principal directions and the second one gives an idea of how different the projected PCA and the $L_2$ PCA are. To assess the interpretability of the principal directions and the scores obtained with the projected PCA, we first compute for every principal direction $\boldsymbol{w}_h^*$ the quantities $\eta_h^{\min}$ and $\eta_h^{\max}$ such that

$$\eta_h^{\min} = \min_{\eta \in \mathbb{R}} \{\boldsymbol{a}_0 + \eta \boldsymbol{w}_h^* \in \mathbb{R}^{J\uparrow}\}$$

where $\boldsymbol{a}_0$ is the spline coefficient vector associated with the barycenter $F_0^-$. The scalar $\eta_h^{\max}$ is found analogously. Hence $(\eta_h^{\min}\boldsymbol{w}_h^*, \eta_h^{\max}\boldsymbol{w}_h^*)$ is the segment spanned by the principal direction living inside the convex cone $\mathbb{R}^{J\uparrow}$. If the scores of all observations

along this direction lie within the range $(\eta_h^{\min}, \eta_h^{\max})$, then the variability captured by (empirical) projected PCA can be decomposed along the principal directions, whose scores are then highly interpretable. Contrary, the PCA scores outside $(\eta_h^{\min}, \eta_h^{\max})$ will be associated with functions which are not quantiles, and thus limiting the interpretability of the direction. Hence, we propose the following *interpretability score*

$$IS_h = 1 - \frac{1}{n}\sum_{i=1}^{n} d\left(s_{ih}, [\eta_h^{\min}, \eta_h^{\max}]\right) / |s_{ih}|, \tag{6.22}$$

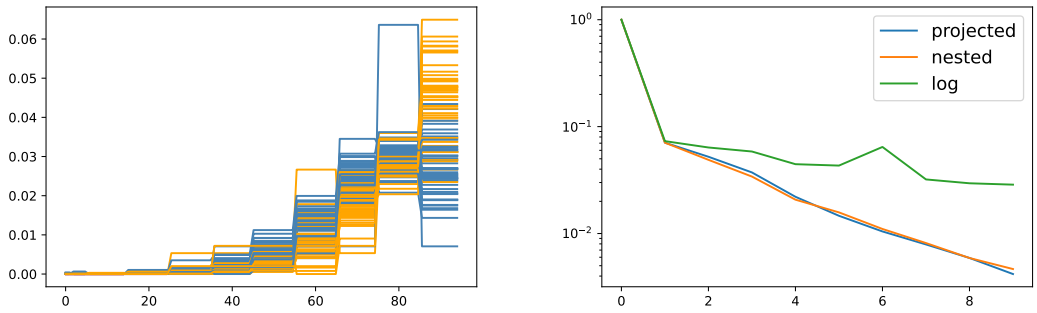
where $s_{ih}$ is the score of observation $i$ along direction $h$ according to the projected PCA. A value of $IS_h$ equal to one corresponds to perfect interpretability, that is, projected PCA behaves like a standard Euclidean PCA along direction $h$. On the other hand, values of $IS_h$ closer to zero indicate that the decomposition of the variance along the principal directions lies outside $\mathbb{R}^{J\uparrow}$ for direction $h$. The interpretability score can be fruitfully used also to evaluate the directions found with the nested PCA, upon replacing the $s_{ih}$'s in (6.22) with the scores given by the nested PCA.

Note that the $IS_h$ score is useful to interpret the directions one at a time. However, it can be the case that some scores along one direction $h'$ lie outside the $(\eta_{h'}^{\min}, \eta_{h'}^{\max})$ range but that the $L_2$ projection on the $h \geqslant h'$ component still lies within the projected component. For instance, this could imply that a projected PC could be similar to a nested one despite having very different directions. A discrepancy between the two can appear when the projections of some data points on the $L_2$ PCA lie outside $\mathbb{R}^{J\uparrow}$. Using the terminology of Proposition 30 this can be measured in terms of difference between the projections $\Pi_k(F^{-*} - F_0^-)$ and $\Pi_{Sp(U_k)\cap(X-x_0)}(F^{-*} - F_0^-) = \Pi_{Sp(U_k)\cap(X-x_0)}(\Pi_k(F^{-*} - F_0^-))$, for a given observation $F^{-*}$. To quantify the loss of information at the level of the component (instead of direction), we propose to measure the "ghost variance" captured by the $L_2$ PCA:

$$GV_k = \frac{1}{n}\sum_{i=1}^{n} \|\Pi_k(F_i^- - F_0^-) - \Pi_{U_X^{F_0^-,k}}(\Pi_k(F_i^- - F_0^-))\|_2 \Big/ \|F_i^- - F_0^-\|_2,$$

that is, the $GV_k$ score measures the quantity of information that is lost due to the projection step or, in other words, the information that we trained our PCA on, but that does not appear in the Wasserstein Space. If $GV_k = 0$ then all the information captured by the $L_2$ PCA is inside the Wasserstein Space, then the projected PCA coincide with the nested one by definition.

Finally, although this situation never occurred in our experience, it might happen that $GV_k$ is small but some $IS'_k$ ($k' \leqslant k$) is large. This means that the subspace identified by the projected PCA is suitable for representing the data, but the single principal directions are not interpretable. In this case, we suggest to take a hybrid approach: use the projected PCA as a fast black-box dimensionality reduction step, thus reducing the dimensionality of each observation from $J$ to $k$, and then use the nested PCA, in dimension $k$, to estimate the directions, the main advantage being the reduction in the computational cost to estimate the nested PCA in this lower dimensional space.




**(a)** **(b)**

**Figure 6.6:** *Left panel: distributions of age at the time of death for Covid-19 patients divided by sex: orange corresponds to female and blue to males. Different lines correspond to different US states / inhabited territories. Right panel: reconstruction error as a function of the dimension of the component for different PCAs. The* $0$*-th principal component is the empirical mean.*

### 6.7.3 Analysis of the Covid-19 mortality data set

We perform PCA analysis on the Covid-19 mortality data publicly available at `data.cdc.gov` as of the first December 2020. The data set collects the total number of deaths due to Covid 19 in the US from January 1st 2020 to the current date, data are subdivided by state, sex, and age. In particular, the ages of the deceased are grouped in eleven bins: $[0, 1), [1, 5), [5, 15), [15, 25), [25, 35), [35, 45), [45, 55), [55, 65), [75, 85), [85, +\infty)$ but we truncate the last bin to $95$ years for numerical convenience. Further, we remove Puerto Rico from the analysis because it presented too many missing values. Our final data set, shown in Figure 6.6(a), consists of 106 samples of the distribution of the ages of patients deceased due to Covid-19, divided by sex and pertaining 53 between US states and inhabited territories.

We apply our usual B-spline approximation with $J = 20$ basis to the quantile functions obtained starting from the histograms in Figure 6.6. This choice of $J$ yields an average approximation error, in terms of Wasserstein distance, of $0.02$. An error this low is to be expected since the quantile functions are piecewise linear functions defined on eleven intervals.

We use this real data set to make a hands-on comparison of the inference that can be obtained employing the projected, nested and log PCA.

We start by comparing the projected and nested PCAs. The first direction found by the nested PCA is identical to the one found by the projected while the second is extremely close: the cosine between the two principal directions is approximately $0.99$. In line with this, the interpretability scores equal $IS_1 = 1$ and $IS_2 \approx 0.89$, while $GV_2 = 0.05$. Moreover, the two-dimensional projected principal component explains

more than $90\%$ of the $L_2$ variability and $NRE_2 \approx 0.05$ for both projected and nested PCA. Given the reconstruction error and the $GV_2$ score, we can conclude that the two-dimensional projected principal component provides a very good fit to the data, and that both selected principal directions are well behaved with respect to their scores, guaranteeing interpretable results.

Considering the discussion above and the fact that both the projected and nested PCA employ metric projection to map data points to the $k$-dimensional principal component, inference obtained with the nested PCA and with the projected one is almost identical in this case and we show results only for the projected PCA in Figure 6.7. In particular, the first principal direction shows that the greatest variability is due to the elders: low negative values along this direction correspond to most of the mortality being concentrated among in the $80+$ range. The red and the green distributions shown in the rightmost panel show two antithetic behaviors which correspond to scores along the first principal direction of roughly $-8.5$ and $7$ as shown in the third panel of Figure 6.7. In fact, the red distribution is concentrated almost exclusively on the last two bins of the histogram, with the $85+$ bin weighting for more of $60\%$ of the deaths. At the opposite, the green distribution gives more weight to lower age values. The second direction instead shows variability in the $40 - 80$ range. The purple distribution, characterized by the highest score along this direction, shows that a significant percentage of deaths occurred in the age range $60 - 75$. Finally, the third panel of Figure 6.7 reports the scores along the first two principal directions for the whole data set, blue dots representing males and orange dots women. We can appreciate how women tend to have lower scores on both directions. This is in line with our understanding that Covid-19 is more severe among the male population (see for instance Lawton, 2020), which explains why males are more susceptible to death even at younger ages, while deaths among women are more concentrated in the $70+$ age range, being the elders in general more fragile.

The comparison with log PCA requires more attention. First of all, note that the directions obtained with the projected and log PCA are the same by definition, since they are both obtained performing PCA in $L_2([0, 1])$, but the principal components may differ because different projection operators are employed when the orthogonal projection of a point onto the principal component lies outside of the image of $\varphi_\mu$, as discussed in Section 6.3.4. As expected from the comparison between the metric projection and the pushforward operator in Figure 6.2, the fit to the data of the projected and log PCAs will be different. In particular, in this case we observe that the log PCA does a worse job in term of $NRE$, as shown in Figure 6.6(b), especially when the dimension increases. This behavior can be also in part explained by the complexity of the numerical routines needed to approximate the pushforward operator (required by the log PCA) where it is natural to expect some numerical errors.

More in general, as discussed also in Cazelles et al. (2018), we can conclude that the log PCA is not suited to study this particular data set because the $L_2$ PCA is different from the nested geodesic PCA (as testified by the $GV_2$ score). In fact, apart from the visual inspection of the $L_2$ principal directions – which are not guaranteed to span the

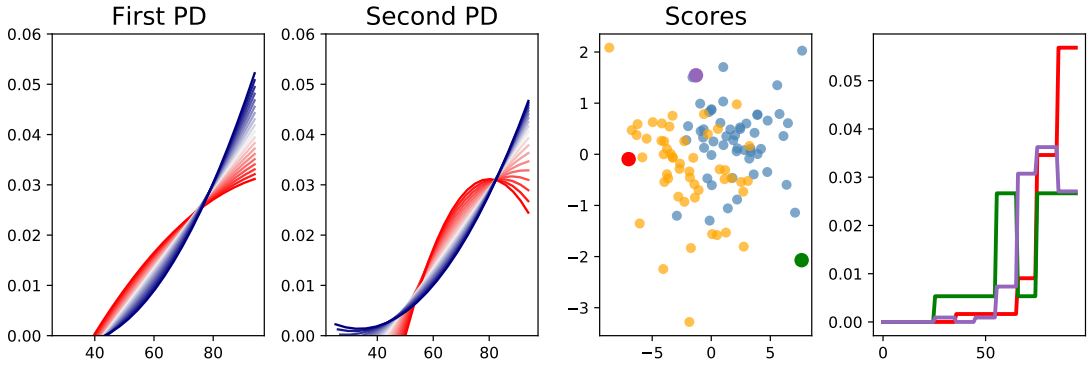



**Figure 6.7:** *The first two panels show the variability along the first two principal directions (first and second panel), using the same visualization technique as in Figure 6.4. The third panel reports the scores of the projections on the two dimensional principal component (orange for women and blue for men) and the fourth panel shows three particular distributions, also highlighted in the third panel. In particular, the red distribution is the one of women in Vermont, the green one are males in Alaska and the purple one are women in West Virginia.*

log-principal components – not much can be obtained from the log PCA in this case, since it does not provide a consistent way of projecting data points on the principal component as pointed out in Section 6.3.4.

## 6.8 Numerical Illustrations for the Distribution on Distribution Regression

In this section, we propose a comparison between the Wasserstein projected and simplicial (see Appendix 6.12) approaches when the task at hand is distribution on distribution regression, and show an application of the Wasserstein projected regression framework to a problem of wind speed forecasting.

### 6.8.1 Simulation Study

We consider two data generating processes as follows. In the first setting, data are generating from the Wasserstein regression: independent variables $z_1, \dots, z_n$ are generated by considering quantile functions $F^-_{z1}, \dots, F^-_{zn}$ such that $F_{zi} = \sum_{h=1}^{30} a^{(z)}_{ih} \psi^{(3)}_j$ where $\psi^{(3)}_1, \dots, \psi^{(3)}_{30}$ is a cubic spline basis over equispaced knots in $[0, 1]$ and $a^{(z)}_{i1} = 0$, $a^{(z)}_{i2} = \delta_{i1}$, $a^{(z)}_{ij} = a^{(z)}_{ij-1} + \delta_{ij-1}$, and $(\delta_{i2}, \dots, \delta_{i30}) \sim$ Dirichlet$(1, \dots, 1)$. This data generating procedure ensures the $F^-_{zi}(0) = 0$, $F^-_{zi}(1) = 1$ and $F^-_{zi}$ is monotonically increasing, cf. Proposition 32. The dependent variables $F^-_{y1}, \dots, F^-_{yn}$ are generated using the same spline expansion of the dependent variables and letting $\boldsymbol{a}^{(y)}_i = B\boldsymbol{a}^{(z)}_i$. $B$

| | First scenario | Second scenario |
|---|---|---|
| Wasserstein | $(4 \times 10^{-7}, 7 \times 10^{-8})$ | $(5 \times 10^{-3}, 6 \times 10^{-3})$ |
| Simplicial | $(0.9, 2.66)$ | $(4 \times 10^{-4}, 5 \times 10^{-4})$ |

**Table 6.1:** *Cross validation (leave one out) errors and standard deviations for the Wasserstein and Simplicial regression under the two simulated examples*

is a randomly generated matrix with rows $\boldsymbol{b}_{\tilde{1}}, \ldots, \boldsymbol{b}_{\tilde{30}}$, and each $\boldsymbol{b}_i$ is generated as follows: $b_{i1} \sim \text{Uniform}(0, 0, 5)$ $b_{ij} = b_{ij-1} + \tilde{b}_{ij}$ and $\tilde{b}_{ij} \sim \text{Uniform}(0, 0, 5)$, so that the coefficients $a_{ij}^{(y)}$ are monotonically non decreasing for each $i$ and thus the $F_{yi}^{-}$'s can be considered quantile functions.

We compute the pushforward of the uniform distribution via numerical inversion and differentiation and obtain the pdf associated to each quantile function. Observe that this task is easier than approximating the pushforward of a generic $\mu$ through a generic $f$ (as Cazelles et al. (2018) do) since the quantile functions are monotonic and we have simple expressions for all the quantities related to $\mu$. Since the simplicial regression takes as input (a transformation of) the pdfs while the Wasserstein regression works directly on the quantile functions, and also due to the fact that numerical errors can be introduced in the data set during the inversion and differentiation, we consider as ground truth the pdfs and, for the Wasserstein approach, re-compute numerically the quantile functions.

In the second setting instead, we generate data from the simplicial regression model: independent variables $z_1, \ldots, z_n$ are generated by applying the inverse of the centered log ratio to a random spline expansion as follows. For each $i = 1, \ldots, n$ let $\tilde{p}_{zi} = \sum_{j=1}^{30} a_{ij}^{(z)} \psi_j^{(3)}$ where the $\psi_j^{(3)}$'s are the same B-spline basis as in the previous setting. Here, the $a_{ij}^{(z)}$'s are generated iid from a Gaussian distribution with mean $0$ and standard deviation $0.2$. The dependent variables are generated by letting $\tilde{p}_{yi} = \sum_{j=1}^{30} a_{ij}^{(y)} \psi_j^{(3)}$ and $\boldsymbol{a}_i^{(y)} = B\boldsymbol{a}_i^{(z)}$, where $B$ is a randomly generated $30 \times 30$ matrix with entries drawn iid from a standard normal distribution. Finally the pdfs $p_{zi}$ ($p_{yi}$) are recovered by applying the inverse of the centered log ratio to $\tilde{p}_{zi}$ ($\tilde{p}_{yi}$), see Appendix 6.12 for more details.

Note that under the second data generating process, both the dependent and independent distributions have support in $[0, 1]$ by construction, whereas under the first data generating process the independent variables might have a larger support. Thus, to fit the simplicial regression in the first scenario, as common practice (cf. Appendix 6.12), we extend the support of all the distributions (both dependent and independent) to the smallest interval of the real line containing all the supports. This is done by adding a small term to the pdfs (in our example, $10^{-12}$) and then renormalizing them.

For both examples, we simulated $100$ observations and compared the projected-Wasserstein and simplicial regression using leave-one-out cross-validation. In particular, for both approaches we use $J = 20$ quadratic spline basis and choose the penalty term $\rho$ in (6.16) through grid search. Table 6.1 shows the pairs of mean squared error and standard deviation of the cross validation, the metric to compare the ground truth

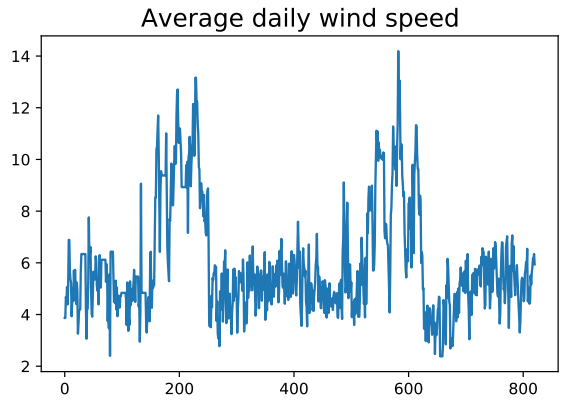



**Figure 6.8:** *Daily average wind speed*

and the prediction is the 2-Wasserstein distance. As one might expect, the Wasserstein regression performs better in the first scenario while the simplicial regression performs better in the second scenario. However, it is surprising how the Wasserstein geometry can capture (in terms of Wasserstein metric) dependence generated by a linear structure which we have shown to be very different from the Wasserstein one, making the projected regression a promising tool for such inferential problems

### 6.8.2 Wind speed distribution forecasting from a set of experts

We consider the problem of forecasting the distribution of the wind speed nearby a wind farm from a set of experts. The data set is publicly available at `www.kaggle.com/theforcecoder/wind-power-forecasting`. In particular, data consists of measurements of the wind speed collected every ten minutes for a period of 821 days starting from the 31st December 2017. The daily average wind speed is shown in Figure 6.8.

We assume to have access to a set of *experts*, that is a set of trained models, that provide a probabilistic one-day-ahead forecast for the average wind speed. Here, our goal is to combine this set of experts and provide a point estimate of the wind speed distribution for the whole day, which can be helpful when planning the maintenance of the wind mills for instance.

Formally, let $K$ denote the number of experts considered, $F^-_{zij}$ is the quantile function associated to the probabilistic forecast of the average wind speed for day $i$ given by expert $j = 1, \dots, K$; $F^-_{yi}$ is the empirical quantile function of the wind speed for day $i$. In particular, we consider $K = 4$ experts built from the *Prophet* model by Facebook (Taylor and Letham, 2018) as follows: model $M1$ is the classical Prophet, without additional covariates or seasonality trends; model $M2$ includes the ambient temperature as covariate but not seasonality; model $M3$ includes a yearly seasonality and no covariates

| | $R1$ | $R2$ | $R3$ | $R2$ | $RF$ |
|---|---|---|---|---|---|
| MSE | $(1.22 \pm 1.32)$ | $(1.19 \pm 1.26)$ | $(1.15 \pm 1.07)$ | $(1.24 \pm 1.23)$ | $(0.86 \pm 0.82)$ |

**Table 6.2:** *Mean square prediction error $\pm$ one standard deviation on the held-out test set.*

and model $M4$ includes both yearly seasonality and ambient temperature as covariate. The models are estimated using variational inference on rolling samples of $365$ days and produce one day ahead probabilistic forecasts for the average wind speed. The final sample size corresponds to $n = 456$.

We consider a trivial extension of the distribution on distribution regression model in Section 6.5.2 as follows:

$$\mathbb{E}[F_{yi}^{-} \mid F_{zi1}^{-}, \dots, F_{ziK}^{-}] = \Pi_{L_2([0,1])^{\uparrow}}\Big(\alpha + \sum_{j=1}^{K} \int_0^1 \beta_j(t,s) F_{zij}^{-}(t)\, \mathrm{dt}\Big) \tag{6.23}$$

Having approximated all the functions through a B-spline expansion, the model reads

$$\mathbb{E}[\boldsymbol{a}_i^{(y)} \mid \boldsymbol{a}_{i1}^{(z)}, \dots, \boldsymbol{a}_{iJ}^{(z)}] = \Pi_{\mathbb{R}^J{}^{\uparrow}}\Big(\boldsymbol{\theta}_\alpha + \sum_{j=1}^{K} \Theta_{\beta_j} E \boldsymbol{a}_{ij}^{(z)}\Big).$$

The procedure for estimating $\boldsymbol{\theta}_\alpha$ and $\Theta_{\beta_1}, \dots \Theta_{\beta_K}$ is analogous to the one outlined in Section 6.5.2.

We compare the prediction performance of five distribution on distribution regression models. Models $R1$ to $R4$ are obtained by fitting model (6.23) using only one of the four experts, $M1$ to $M4$, while the fifth model ($RF$) is the "full" model in (6.23) considering all the four experts. For this comparison, we perform a train-test split of the $456$ days for which the experts produced the prediction, considering the last $100$ days as test. We select hyperparameters (namely, the penalty coefficient $\rho$ in (6.16) and whether to include or not the intercept term $\alpha$) by a grid search cross validation on the training set, and compare the mean square error on the held-out test set. Results of the comparison are reported in Table 6.2. As expected, the model with the four predictors ($RF$) is the best performer. Interestingly, all the other models $R1$-$R4$ perform similarly and present a much higher mean square error when compared to $RF$, thus suggesting that the best performance is achieved by combining the different experts together and no expert alone can be a good predictor. This is possibly explained by some experts being able to better forecast one scenario (for instance, light winds) and other experts being able to better forecast other scenarios.

We conclude with some descriptive analysis. Figure 6.9 shows the point estimates for the coefficients $\beta_j$. We can interpret as highly influential for the regression the areas of the $\beta_j$'s with high absolute value, and as negligible area with values close to zero.

We can highlight some differences among the coefficients in Figure 6.9. In particular, model $M1$, seems influent when predicting the tails of the distribution, in particular

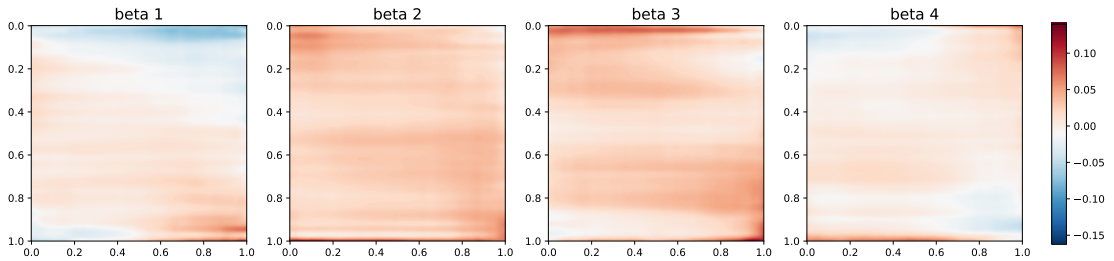



**Figure 6.9:** *Estimates of the $\beta_i(t,s)$'s evaluated on $[0,1]^2$. The variable $t$ runs across columns, and variable $s$ across rows*

with negative weights for the left tail and positive weights for the right tail. Model $M2$, seems to be affecting all the steps of the prediction and in particular to be model affecting the most the median of the distribution. Model $M3$, appears to be, with $M2$, the most important model for the prediction: the absolute value in the corresponding regressor $\beta_3$ is often very high and with noticeable peaks corresponding to areas predicting the left tail and towards the right tail. Finally, the regressor corresponding to $M4$ has very low values thus resulting in minor importance in terms of regression influence.

Interestingly, the experts providing the most precious inputs to our regression model are $M2$ and $M3$, that incorporate only the seasonality effect and the temperature covariate respectively, while $M4$, which incorporates both, seems to be less important. Hence, the regression model in (6.23) finds more effective combining experts trained on different covariates than correcting an expert already trained on all the covariates. In particular, our insight is that $M2$ is responsible for centering the median of the output distribution. The tails of the distribution seem to need also the contribution of seasonality data, given by $M3$. Finally, we also observe that the left tail of the wind distribution seems the most difficult to be predicted, needing very high positive and negative weights across different models, to be obtained.

## 6.9 Discussion and Further Directions

In this chapter, we propose a novel class of *projected* statistical methods for distributional data on the real line, focusing in particular on the definition of a *projected* PCA and a *projected* linear regression. By investigating the weak Riemannian structure of the Wasserstein space and the transport maps between probability measures, we represent the Wasserstein space as a closed convex cone inside an Hilbert space.

Similar to *log* methods, our models exploit the possibility to map data into a linear space to perform statistics in an *extrinsic* fashion. However, instead of using operators like the *exp* map or a some kind of boundary projection to return to the Wasserstein space, we rely on a metric projection operator that is more respectful of the underlying

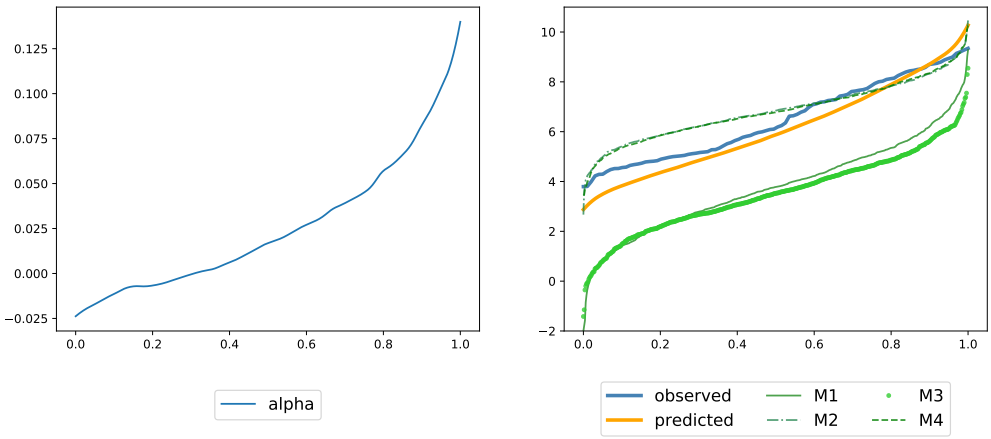



**Figure 6.10:** *Estimate of $\alpha$ (left) and prediction of one $F_y^-$ of the test set (right). In the right panel, the blue line corresponds to the empirical quantile function, the orange one to the prediction from $RF$ and the green ones to the average wind predictions obtained from the experts $M1$-$M4$.*

metric.

By choosing as base point the uniform measure on $[0, 1]$, we are able to efficiently approximate the metric projection operator so that our models combine the ease of implementation of *extrinsic* methods while retaining a performance similar to the one of *intrinsic* methods. Further, through a quadratic B-spline approximation, we can greatly reduce the dimensionality of the optimization problems involved, resulting in fast empirical methods. As a byproduct of this approach, we also derive faster numerical routines for the *geodesic* PCA in Bigot et al. (2017).

We study asymptotic properties of the proposed methods, concluding that, under reasonable regularity assumptions, our *projected* models provide consistent estimates and that the B-spline approximation error becomes negligible. We showcase our approach in several simulation studies and using two real world data sets, comparing our models to *intrinsic* and *extrinsic* ones and to the *simplicial* approach in Hron et al. (2014), concluding that the *projected* PCA and regression constitute a valid candidate for performing inference on a data set of distributions.

Although our *projected* framework was proven to be viable in many practical situations, some care must be taken when adopting it, especially when performing PCA. In fact, the *extrinsic* nature of our method might not fit every data set, in which case a more computationally demanding *intrinsic* PCA might be preferred, see for instance Appendix 6.14.1 for an example where the *projected* principal directions are not interpretable. On top of that, performing PCA in the Wasserstein space requires more attention than performing the usual Euclidean PCA: as pointed out in Appendix 6.14.2, since principal components are not linear subspaces, decomposing the variance along the directions (i.e., looking at the scores) must be done carefully, and making sure that

the directions are indeed interpretable. To assist practitioners, in Section 6.7.2 we have also proposed two scores that quantify the interpretability of the principal directions and the discrepancy between the *nested* and *projected* principal components.

Several extensions and modifications of our approach are possible. One possibility is to extend our framework to encompass more models, such as generalized linear models and independent component analysis. Although this should be straightforward in theory, the numerical computations could become more burdensome. Furthermore, as an alternative to our approach based on B-splines approximation, one could use such B-spline expansion only to approximate the metric projection operator. Another interesting line of research would consist in building hybrid approaches (as anticipated in Section 6.7.2) to analyze distributions in the Wasserstein space, using both *extrinsic* and *intrinsic* methods to exploit the advantages of both worlds, while mitigating the disadvantages. We also think that a deeper comparison between the Wasserstein and the simplicial geometries could help practitioners in choosing between them.

Finally, as pointed out by an anonymous referee, extensions to encompass measures supported on $\mathbb{R}^d$, $d > 1$, are of great interest. This is surely a very challenging problem, due to the geometric structure of $\mathcal{W}_2(\mathbb{R}^d)$. We identify three main obstacles in this sense. First, the map onto the tangent space is not an isometry because the Wasserstein space is curved. Second, we lose the nice characterization of the tangent space and of the image of $\log_\mu$, so that the metric projection operator becomes harder to derive. Third, the computational cost would greatly increase due to the need of numerically approximating the transport maps needed to compute the distances.

## 6.10 Acknoweldgements

We thank two anonymous referees for their detailed and helpful reviews, which helped us improving the quality and the clarity of this work. We also thank Riccardo Scimone for helpful feedback and comments on an earlier draft of this work and Federico Bassetti, Alessandra Guglielmi helpful for discussions.

## 6.11 Proofs

*Assumptions on $x_0$.*

Let $B_\varepsilon(x_0) = \{x \in H \,\big|\, ||x - x_0|| < \varepsilon\}$, a ball of radius $\varepsilon$ in $H$. Given a set $C$, we refer to aff$(C)$ as the smallest affine subset containing $C$, found as the intersection of all affine subspaces containing $C$. Similarly $\mathcal{H}(C)$ is the convex hull of $C$, the smallest convex subset of $H$ containing it. The relative interior of a set $C$ is defined as its interior considering as ambient space aff$(C)$: relint$(C) = \{x \in C \,\big|\, \exists B_\varepsilon(x_0)$ such that $B_\varepsilon(x_0) \cap$ aff$(C) \subset C\}$.

Throughout our paper we assume that the random variable $\mathcal{X}$ is such that (i) there exists $x_0 = \mathbb{E}[\mathcal{X}]$ and (ii) $x_0 \in \text{relint}(\mathcal{H}(\text{supp}(\mathcal{X})))$ where $\text{supp}(\mathcal{X})$ is the support of $\mathcal{X}$. These assumptions are indeed quite natural and require that the distribution of $\mathcal{X}$ has a well defined barycenter, which is not in a "degenerate" position with respect to the convex hull of its support, which may happen in infinite dimensional Hilbert Spaces. See, for instance, Berezin and Miftakhov (2019) for an example of distributions not verifying this second assumption.

*Proof of* Lemma 12.

The proof is divided in two steps. First, we prove that $(x_0 + Sp(U_k)) \cap X$ has dimension $k$. Then, we show that $U_X^{x_0,k} = (x_0 + Sp(U_k)) \cap X$. Without loss of generality, for ease of notation, we perform an affine change of variable so that $x_0 = 0$, but, with a slight abuse of notation, we keep denoting with $\mathcal{X}$ and $X$ the transformed random variable and the convex cone respectively.

To prove the first part, let $\mathcal{H}(\mathcal{X})$ be the convex hull of the support of $\mathcal{X}$ and $\text{aff}(\mathcal{H}(\mathcal{X})) = K$ be the smallest affine subset of $H$ containing $\mathcal{H}(\mathcal{X})$. We know by assumption that there is an open ball in $K$ which contains $x_0 = 0$ and is contained in $\mathcal{H}(\mathcal{X})$. Moreover, for every $k \leqslant dim(K)$, $Sp(U_k) \subset K$. Note that we can clearly suppose $k \leqslant dim(K)$, otherwise principal components analysis is useless. With this assumption, since $x_0 = 0$ is in the relative intern of $\mathcal{H}(\mathcal{X})$, we have $k = dim(Sp(U_k) \cap \mathcal{H}(\mathcal{X})) \leqslant dim(Sp(U_k) \cap X) \leqslant k$.

Now we prove that a $(k, 0)$-projected principal component is given by $Sp(U_k) \cap X$. To prove this, let $C^*$ be a $(k, 0)$-projected principal component and $A^* = A \cap X$, with $A = Sp(U_k)$: we know (i) $x_0 = 0 \in A^*$, (ii) $dim(A^*) = k$ by definition and (iii) $A^* \subseteq \Pi_X(A)$, so we have $A^* \subset C^*$.

Since $dim(C^*) = k$ there is $C$ linear subspace of dimension $k$ such that $C^* \subset C$. Consider $C' = C \cap X$: clearly $C^* \subset C'$, so that $A^* \subset C^* \subset C'$. Moreover, $A^* \subset C'$, which implies $A \cap X \subset C \cap X$ and thus $Sp(A \cap X) \subset Sp(C \cap X)$. The proof is concluded if $dim(Sp(A \cap X)) = dim(Sp(C \cap X)) = k$. In fact, in this case $A = Sp(A \cap X)$ and $C = Sp(C \cap X)$ which means that $A \subset C$ and since $dim(A) = dim(C) = k$, $A$ and $C$ coincide, proving $A^* = C^*$.

To prove this final claim, observe that $dim(Sp(A \cap X)) < k$ implies $dim(A \cap X) < k$, which contradicts the proof of the first part of this Lemma. Similarly, $dim(Sp(C \cap X)) = k$ since $dim(C^*) = k$ by hypothesis.

■

*Proof of* Proposition 29.

The fact that $\|\Pi_{U_X^{x_0,k}}(x) - x\| \geqslant \|\Pi_{U_X^{x_0,k+1}}(x) - x\|$ follows easily by noticing that $U_X^{x_0,k} \subset U_X^{x_0,k+1}$.

Now, to prove that $\|\Pi_{U_X^{x_0,k}}(x) - x\| \to 0$ as $k$ increases, we first notice that, by the properties of the principal components in $H$ we have $\Pi_{Sp(U_k)}(x - x_0) \xrightarrow{k} x - x_0$ for every $x \in X$, which implies $\|\Pi_{Sp(U_k)+x_0}(x) - x\| \to 0$. Then, denote $x_1 = \Pi_{U_X^{x_0,1}}(x)$

and let $r_k$ be the line between $x_1$ and $x$. Let:

$$x_k = \underset{x' \in r_k \cap Sp(U_k)+x_0}{\arg\min} \|x' - x\|$$

We clearly have have $x_k \to x$. Finally, by convexity we know $x_k \in U_X^{x_0,k}$, which implies $\|\Pi_{U_X^{x_0,k}}(x) - x\| \leqslant \|x_k - x\| \to 0$.

■

*Proof of* Proposition 30.

Again, without loss of generality, for ease of notation, we perform an affine change of variable so that $x_0 = 0$, but, with a slight abuse of notation, we keep denoting with $\mathcal{X}$ and $X$ the transformed random variable and convex cone respectively.

We start by noticing that being $\Pi_k$ the orthogonal projection onto a subspace, $x - \Pi_k(x) \perp Span(U_k)$ and thus for $v \in Span(U_k)$:

$$\|x^* - v\|^2 = \|x^* - \Pi_k(x^*)\|^2 + \|\Pi_k(x^*) - v\|^2$$

Then

$$\underset{v \in U_X^{0,k}}{\arg\min} \|x^* - v\| = \underset{v \in Sp(U_k) \cap X}{\arg\min} \|\Pi_k(x^*) - v\|$$

and the result follows. ■

*Proof of* Proposition 32.

1. As shown in the supplementary of Pya and Wood (2015) by standard B-spline formulas we obtain that given $f(x) = \sum_{j=1}^{J} a_j \psi_j^k(x)$, then $f'(x) = \sum_{j=1}^{J} (a_j - a_{j-1}) \cdot \psi_j^{k-1}(x)$. Being the B-spline basis function nonnegative by definition, we obtain the result.

2. With $k = 2$, $f'(x)$ on the interval $[x_{j+1}, x_j]$ has the following expression:

$$\frac{x - x_j}{x_{j+1} - x_j} \cdot (\alpha_j - \alpha_{j-1}) + \frac{x_{j+1} - x}{x_{j+1} - x_j} \cdot (\alpha_{j-1} - \alpha_{j-2})$$

so:

$$lim_{x \to x_{j+1}^-} f'(x) = \alpha_j - \alpha_{j-1}$$

and the result follows. ■

*Proof of* Proposition 33 and 34.

We report here Propositions 3.3 and 3.4 of Bigot et al. (2017), with the notation adapted to our manuscript. In the following $H$ is a separable Hilbert space, $X$ is a closed convex subset of $H$, $\mathcal{X}$ is an $X$-valued square integrable random variable, $x_0$ a point in $X$ and $k \geqslant 1$ an integer.

**Proposition 38.** *Let $U^* = \{u_1^*, .., u_k^*\}$ be a minimizer over orthonormal sets $U$ of $H$ of cardinality $k$, of $D_X^{x_0}(\mathcal{X}, U) := \mathbb{E}d^2(\mathcal{X}, (x_0 + Sp(U)) \cap X)$, then $U_X^{x_0} := (x_0 + Sp(U)) \cap X$ is a $(k, x_0)-$global principal component of $\mathcal{X}$.*

**Proposition 39.** *Let $U^* = \{u_1^*, .., u_k^*\}$ be an orthonormal set such that $U_i^* = \{u_1^*, .., u_i^*\}$ is a minimimizer of $D_X^{x_0}(\mathcal{X}, U)$ over the orthonormal sets of cardinality "$i$" such that $U \supset U_{i-1}^*$; then $U_X^{*x_0}$ is a $(k, x_0)-$nested principal convex component of $\mathcal{X}$.*

Applying Propositions 38 and 39 we can obtain equivalent definitions of geodesic and nested PCA as optimization problems in $L_2([0,1])$. If we fix $J \in \mathbb{N} > 0$ and a quadratic B-spline basis $\{\psi_j\}_{j=1}^J$, we can use Propositions 38 and 39 with $X = L_2([0,1])^{J\uparrow}$ and $H = L_2([0,1])^J$. Thanks to Remark 27 we obtain the results. ■

*Proof of* Proposition 35.

Let $S_J = \sum_{j=1}^J \lambda_j^{(J)} \psi_j^{(J)}$ and its derivative $s_J = \sum_j (\lambda_j^{(J)} - \lambda_{j-1}^{(J)}) \widetilde{\psi}_j^{(J)}$ where $\widetilde{\psi}_j^{(J)}$ denotes the linear spline basis on the same equispaced grid in $[0,1]$.

Let $f_\mu^- = (F_\mu^-)'$, of course it can be seen that $f_\mu^-$ is non-negative. Moreover, it is obvious that $f_\mu^- \in W_2^\infty([0,1])$. Then, from De Boor and Daniel (1974b) we get that there exist $s_J$ such that $\|s_J - f_\mu^-\|_\infty \leqslant C\|D^2 f_\mu^-\|_\infty J^{-2}$, where $C$ is a constant depending on the interval $[0,1]$ but not on $n$.

Hence, we can determine the coefficients $\{\lambda_j^{(J)}\}$, starting from the spline $s_J$, up to a translation factor.

We fix a particular set of coefficients by letting $S_J(0) = \lambda_1^{(J)} = F_\mu^-(0)$ for each $J$. So that:

$$S_J(x) - F_\mu^-(x) = \int_0^x s_J(t)dt - \int_0^x f_\mu^-(t)dt - S_J(0) + F_\mu^-(0) = \int_0^x s_J(t) - f_\mu^-(t)dt$$

By using the previous result, the integral we have that $S_J(x) - F_\mu^-(x) \leqslant CJ^{-2}$ for all $x$ which proves the proposition. ■

*Proof of* Proposition 36.

By the Assumptions in Section 6.6.2 and Remark 30 there exists a ball $B_K$ in $W_3^\infty([0,1])$ of radius $K$ for some $K > 0$, such that each $F_i^-$ can be $\varepsilon$-approximated by $\widetilde{F}_i^- \in W_3^\infty([0,1])$ with $\widetilde{F}^- \in B_K$. We can suppose that also the eigenvectors of the covariance operator of the generating process belong to such sphere, otherwise we just increase its radius of some finite amount.

By Proposition 35 we can choose a spline basis (that is, a number of elements $J > 0$), such that we get a $\varepsilon$-uniformly good approximation of $B_K$ (and thus we can $2\varepsilon$-approximate its $L_2$ closure). To lighten notation, thanks to Remark 27 we deliberately confuse $\mathbb{R}^{J\uparrow}$ and the space monotone $B$-splines with $J$ basis functions, the inner product we are referring to will always be clear by looking at its entries.

Now consider the following inequalities, with $\boldsymbol{a}_i^J$ obtained as $2\varepsilon$ approximations of $F_i^-$, $\boldsymbol{w}^J \in \mathbb{R}^{\mathbb{J}}$, $w \in L_2([0,1])$:

$$\Big|\frac{1}{n}\sum_i \langle F_i^-, w\rangle^2 - \frac{1}{n}\sum_i \langle \boldsymbol{a}_i^J, \boldsymbol{w}^J\rangle^2\Big| \leqslant \frac{1}{n}\Big|\sum_i \langle F_i^-, w\rangle^2 - \sum_i \langle \boldsymbol{a}_i^J, w\rangle^2 + \sum_i \langle \boldsymbol{a}_i^J, w\rangle^2 - \sum_i \langle \boldsymbol{a}_i^J, \boldsymbol{w}\rangle^2\Big|,$$

where the inner product $\langle \boldsymbol{a}_i^J, w\rangle$ is to be intended as the $L_2$ inner product between the spline function with coefficients $\boldsymbol{a}_i^J$ and the $L_2$ function $w$. Consider now:

$$\begin{aligned}
&\frac{1}{n}\sum_i (\langle F_i^-, w\rangle^2 - \langle \boldsymbol{a}_i^J, w\rangle^2) = \\
&\qquad \frac{1}{n}\sum_i (\langle F_i^-, w\rangle - \langle \boldsymbol{a}_i^J, w\rangle)(\langle F_i^-, w\rangle + \langle \boldsymbol{a}_i^J, w\rangle) = \\
&\qquad \frac{1}{n}\sum_i \langle F_i^- - \boldsymbol{a}_i^J, w\rangle\langle F_i^- + \boldsymbol{a}_i^J, w\rangle \leqslant \\
&\qquad \frac{1}{n}\sum_i \Big|\langle F_i^- - \boldsymbol{a}_i^J, w\rangle\Big| \cdot \Big|\langle F_i^- + \boldsymbol{a}_i^J, w\rangle\Big| \leqslant \\
&\qquad \frac{1}{n}\sum_i 2\varepsilon\|w\|^2 2K = 4\varepsilon K\|w\|^2
\end{aligned}$$

Similarly:

$$\Big|\frac{1}{n}\sum_i (\langle \boldsymbol{a}_i^J, w\rangle^2 - \langle \boldsymbol{a}_i^J, \boldsymbol{w}^J\rangle^2)\Big| \leqslant \|\boldsymbol{a}_i^J\|^2 \cdot \|w - \boldsymbol{w}^J\| \cdot (\|w\| + \|\boldsymbol{w}^J\|)$$

We know that a solution to the problem $\max_{\|w\|_{L_2}=1} \frac{1}{n}\sum_i \langle F_i^-, w\rangle^2$ is given by the first eigenfunction $\widehat{w}$ of the covariance operator of the empirical process. Now we are in the condition to apply results in Dauxois et al. (1982), or in Qi and Zhao (2011) (with $\alpha \to 0$) to conclude that $\widehat{w}$ converges to the first eigenfunction $\bar{w}$ of the covariance operator of the process that generates $F_i^-$. By hypothesis, such eigenfunction $\bar{w}$ lies in $B_K$ and thus can be approximated with our fixed spline basis. Thus for high enough $n$, also $\widehat{w}$ can be approximated up to $2\varepsilon$.

Let $\boldsymbol{a}_{\widehat{w}}$ be the coefficients of the spline expansion of $\widehat{w}$ spline approximation, that is, $\|w - \boldsymbol{a}_w\| \leqslant 2\varepsilon$. Observe that $\Big|\|\widehat{w}\|_2 - \|\boldsymbol{a}_{\widehat{w}}\|_E\Big| \leqslant 2\varepsilon$, just as $\|\boldsymbol{a}_J^i\| \leqslant K + 2\varepsilon$. Thus, up to adding another $\varepsilon$ to the approximation error $\|\widehat{w} - \boldsymbol{a}_{\widehat{w}}\|$, we can suppose $\|\boldsymbol{a}_{\widehat{w}}\|_2 = 1$. Hence:

$$\Big|\frac{1}{n}\sum_i (\langle \boldsymbol{a}_i^J, \widehat{w}\rangle^2 - \langle \boldsymbol{a}_i^J, \boldsymbol{a}_{\widehat{w}}\rangle^2)\Big| \leqslant (K + 2\varepsilon) \cdot 3\varepsilon \cdot 2$$

Which leads to:

$$\Big| \max_{\|w\|_{L_2}=1} \sum_i \langle \boldsymbol{a}_i^J, w\rangle^2 - \max_{\|\boldsymbol{w}^J\|_E=1} \sum_i \langle \boldsymbol{a}_i^J, \boldsymbol{w}^J\rangle^2 \Big| \leqslant (K+2\varepsilon)\cdot 3\varepsilon \cdot 2$$

Finally, combining the above results and the fact that $|\max f - \max g| \leqslant \max |f-g|$ for any pair of real valued functions $f$ and $g$, we obtain:

$$\Big| \max_{\|w\|_{L_2}=1} \frac{1}{n}\sum_i \langle f_i, w\rangle^2 - \max_{\|\boldsymbol{w}^J\|_E=1} \frac{1}{n}\sum_i \langle \boldsymbol{a}_i^J, \boldsymbol{w}^J\rangle^2 \Big| \leqslant$$
$$\max_{\|w\|_{L_2}=1} 4\varepsilon K\|w\| + (K+2\varepsilon)\cdot 6\varepsilon \leqslant 6\varepsilon K(1+2\varepsilon)$$

Thus for instance if we ask that $\varepsilon < 1$, we obtain the desired result with $D = 18\cdot K$. Consistency follows since $\|\boldsymbol{a}_{\widehat{w}} - \bar{w}\| \leqslant \|\boldsymbol{a}_{\widehat{w}} - \widehat{w}\| + \|\widehat{w} - \bar{w}\|$. ■

*Proof of* Lemma 13.

Since for any $x \in X$ we have $\Pi_{\mathbb{R}^{J\uparrow}}(x) \to x$, for any $v \in H$:

$$\|v - \Pi_{\mathbb{R}^{J\uparrow}}(v)\| \leqslant \|v - \Pi_{\mathbb{R}^{J\uparrow}}(\Pi_X(v))\| \leqslant \|v - \Pi_X(v)\| + \|\Pi_X(v) - \Pi_{\mathbb{R}^{J\uparrow}}(\Pi_X(v))\|$$

which implies $\Pi_{\mathbb{R}^{J\uparrow}}(v) \to \Pi_X(v)$. Consider now $\beta_n \to \beta$ in $H$; we have the inequality:

$$\|\Pi_{\mathbb{R}^{J\uparrow}}(\beta_n) - \Pi(\beta)\| \leqslant \|\Pi_{\mathbb{R}^{J\uparrow}}(\beta_n) - \Pi_X(\beta_n)\| + \|\Pi_X(\beta_n) - \Pi_X(\beta)\|$$

the first term of the right hand side of the inequality can be sent to 0 by increasing $J$, the other by increasing $n$. ■

*Proof of* Proposition 37.

We call $a_i$ the spline coefficients associated to $x_i$ and $b_i$ the ones associated to $y_i$. Again we deliberately confuse the spaces where the coefficients live to lighten the notation. Since the penalty term does not depend on the data, we have:

$$\frac{1}{n}\Big| \sum_i \|y_i - \langle x_i, B^T AB\rangle\|^2 - \sum_i \|b_i - \langle a_i, B^T AB\rangle_{L_2([0,1])}\|^2 \Big| =$$
$$\frac{1}{n}\Big| \sum_i (\|y_i - \langle x_i, B^T AB\rangle\|^2 - \|b_i - \langle a_i, B^T AB\rangle_{L_2([0,1])}\|^2) \Big| \leqslant$$
$$\frac{1}{n}\sum_i \|y_i - \langle x_i, B^T AB\rangle\|^2 - \|b_i - \langle a_i, B^T AB\rangle_{L_2([0,1])}\|^2 \Big|$$

Now, since

$$\Big| \|y_i - \langle x_i, B^T AB\rangle\|^2 - \|b_i - \langle a_i, B^T AB\rangle_{L_2([0,1])}\|^2 \Big| =$$
$$\Big| (\|y_i - \langle x_i, B^T AB\rangle\| - \|b_i - \langle a_i, B^T AB\rangle\|) \times$$
$$(\|y_i - \langle x_i, B^T AB\rangle\| + \|b_i - \langle a_i, B^T AB\rangle\|) \Big|$$

Then for some constant $K$ depending on the bounds in the Assumptions, we get:

$$\begin{aligned}\Big|\|y_i - \langle x_i, B^T AB\rangle\|^2 - \|b_i - \langle a_i, B^T AB\rangle_{L_2([0,1])}\|^2\Big| \leqslant \\ \|y_i - \langle x_i, B^T AB\rangle - b_i + \langle a_i, B^T AB\rangle\| 2K = \\ \big(\|y_i - b_i\| + \langle a_i - x_i, B^T AB\rangle\big)\, 2K\end{aligned}$$

Thus, if $J$ is such that we have $\varepsilon$-approximations of the data, by Cauchy-Schwartz we obtain:

$$\frac{1}{n}\Big|\sum_i \|y_i - \langle x_i, B^T AB\rangle\|^2 - \sum_i \|b_i - \langle a_i, B^T AB\rangle_{L_2([0,1])}\|^2\Big| \leqslant K' \cdot \varepsilon$$

for some $K'$ constant.

Thanks to the results in Prchal and Sarda (2007), for any $\varepsilon > 0$, if the number of samples is big, $\widehat{\Theta}$ and $\widehat{\Theta}_J$ exist with probability $1 - \varepsilon$ and are unique. Since the value of the minimization problem the solve are arbitrarily close, then the minimizers converge in $\mathbb{R}^{J\times J}$ with the metric given by the spline basis. ■

*Strong convergence implies semi-norm convergence.*
Let $\mathcal{Z}$ be an $H$-valued random variable and $\mathcal{C}_\mathcal{Z}$ the covariance operator associated to $\mathcal{Z}$, that is:

$$(\mathcal{C}_\mathcal{Z} f)(s) = \int_{[0,1]} cov(\boldsymbol{x}(s), \boldsymbol{x}(t)) f(t) dt.$$

In the following, we denote with $\|\cdot\|_{L_2}$ the $L_2([0,1]^2)$ norm. Further, recall that $\|cov(\mathcal{Z}(s), \mathcal{Z}(t))\|_{L_2([0,1]^2)} = \mathbb{E}[\|\mathcal{Z}\|^2]$. We want to look at the behavior of $\|\widehat{\beta}_{\text{PS}} - \widehat{\beta}_J\|_{\mathcal{C}_\mathcal{Z}}$.

$$\begin{aligned}\int_{[0,1]} \langle \mathcal{C}_\mathcal{Z}(\widehat{\beta}_{\text{PS}}(s,t) - \widehat{\beta}_J(s,t)), \widehat{\beta}_{\text{PS}}(s,t) - \widehat{\beta}_J(s,t)\rangle dt \leqslant \\ \|\mathcal{C}_\mathcal{Z}(\widehat{\beta}_{\text{PS}}(s,t) - \widehat{\beta}_J(s,t))\|_{L_2} \cdot \|\widehat{\beta}_{\text{PS}}(s,t) - \widehat{\beta}_J(s,t)\|_{L_2} \leqslant \\ \mathbb{E}[\|\boldsymbol{x}\|^2] \cdot \|\widehat{\beta}_{\text{PS}}(s,t) - \widehat{\beta}_J(s,t)\|_{L_2} \cdot \|\widehat{\beta}_{\text{PS}}(s,t) - \widehat{\beta}_J(s,t)\|_{L_2}.\end{aligned}$$

So $\|\widehat{\beta}_{\text{PS}} - \widehat{\beta}_J\|_{\mathcal{C}_\mathcal{Z}} \leqslant M \cdot \|\widehat{\beta}_{\text{PS}} - \widehat{\beta}_J\|^2_{L_2}$ for some constant $M$. Thus $\|\cdot\|_{L_2}$ convergence implies $\|\cdot\|_{\mathcal{C}_\mathcal{Z}}$ convergence.

## 6.12 The simplicial approach

The simplicial approach to distributional data analysis is based on the definition of Bayes space $\mathcal{B}^2(I)$ (Egozcue et al., 2006). Formally, let $I \subset \mathbb{R}$ a closed interval, the Bayes spaces $\mathcal{B}^2(I)$ is defined the equivalence class of probability densities $p(x)$ on $I$ (that is $p(x) \geqslant 0$ and $\int_I p(x)dx = 1$) with square integrable logarithm.

The Bayes space is endowed with a linear space starting from the definition of the perturbation and powering operators, that are analogous to the sum and multiplication times a scalar, and inner product. Moreover Menafoglio et al. (2014) defines an isometric isomorphism between $\mathcal{B}^2(I)$ and $L_2([0,1])$ through the so-called centered log ratio (clr) map defined as

$$\widetilde{p}(x) := \text{clr}(p)(x) = \log(p(x)) - \frac{1}{b-a}\int_a^b \log p(t)dt \tag{6.24}$$

for every $p \in \mathcal{B}^2(I)$. The inverse map is defined as

$$p(x) = \text{clr}^{-1}(\widetilde{p})(x) = \frac{\exp(\widetilde{p}(x))}{\int_I \exp(\widetilde{p}(x))dx}$$

Thus, it is possible to define a *simplicial* PCA and *simplicial* regression on the Bayes space starting from the clr map. In particular, let $p_1, \dots, p_n$ be observed densities on the interval $I$ and let $\widetilde{p}_i = \text{clr}(p_i)$. Denote with $\widetilde{w}_1, \dots, \widetilde{w}_k$ the first $k$ principal directions estimated from the $\widetilde{p}_i$'s, then a $k$ dimensional simplicial principal component is the span of $\{w_i = \text{clr}^{-1}(\widetilde{w}_i)\}_{i=1}^k$ in $\mathcal{B}^2(I)$.

Similarly, for pdfs $\{(p_z, p_y)_i\}_{i=1}^n$ a simplicial regression model is defined starting from the clr transformed variables. Let $\widetilde{\Gamma}$ denote a functional regression model in $L_2$ for variables $\{(\widetilde{p}_z, \widetilde{p}_y)_i\}_{i=1}^n$, then the simplicial regression states:

$$\mathbb{E}[p_{yi} \mid p_{zi}] = \text{clr}^{-1}\left(\widetilde{\Gamma}(\widetilde{p}_{zi})\right).$$

Apart from the different geometries of the Wasserstein and Bayes space, which are discussed in Sections 6.7 and 6.8, we can highlight one particular drawback from the simplicial approach, which we believe poses a significant limit to its usefulness. In fact, the main assumption is that all the pdfs $p_i$ share the same support, which might not be the case (for instance, it is not the case for our example in Section 6.8.2). In practice, one may circumvent this need by either "padding" all the pdfs to the same support, i.e considering

$$\overline{p}_i(x) \propto p_i(x) + \varepsilon \mathbb{I}[x \in I], \tag{6.25}$$

where $\mathbb{I}[\cdot]$ denotes the indicator function, and the proportionality is due to the need of re-normalizing the $\overline{p}_i$'s so that they integrate to 1. Another approach could consist in considering $I$ as the intersection of all the supports of the different $p_i$'s let truncate all the pdfs to the shared interval $I$.

Both approaches present undesired side effects that can greatly alter the results. The second approach might end up with a very small interval $I$, so that a lot of information is lost due to this pre-processing step. The drawback of the first approach instead is due to numerical instability. In fact, one would like $\varepsilon$ in (6.25) to be small in order not to corrupt the true signal, given by $p_i$. However, considering the transformation in (6.24) having a small $\varepsilon$ would cause the $\widetilde{p}_i$ to present some extreme values (negative) in

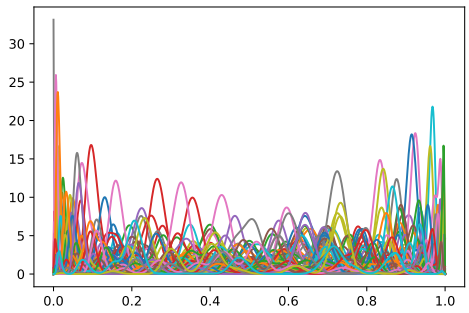



**Figure 6.11:** *Example of data set from* (6.26)

correspondence to $\varepsilon$. Performing PCA on a data set processed in this way would greatly alter the results, as most of the variability of the $\widetilde{p}_i$'s would be masked by a difference in their support.

## 6.13 Additional Simulations

### 6.13.1 Sensitivity Analysis to the Number of Basis Functions

In this simulation, we show how the number of B-spline basis functions affects the inference in our projected PCA and in the simplicial one. In this Scenario, the probability measures are simulated as mixture of beta densities, also known as Bernstein polynomials, as follows:

$$\begin{aligned} p_i(x) &= \sum_{j=1}^{K} w_{ij}\beta(x; j, K-j) \\ \boldsymbol{w}_i &\sim \text{Dirichlet}_K(0.01) \end{aligned} \tag{6.26}$$

Where $\beta(x; a, b)$ denotes the density of a beta distributed random variable with parameters $(a, b)$ evaluated in $x$. By definition, the $p_i$s generated from (6.26) have a fixed support $I = [0, 1]$. See Figure 6.11.

In this setting instead, we let $\mu_i$ in (6.20) be the probability measure associated to $p_i$ and not its smoothed version. Hence, in addition to the amount of information lost during the PCA another factor comes into play: the amount of information that is lost due to the B-spline representation.

Figure 6.12 shows the results. We can see that the reconstruction errors decrease when the dimension of the principal component increases both for the simplicial and projected PCA. Moreover, as the number of B-spline basis increase, the performance tend to get a little bit worse for both the approaches. We believe that this is due to an increased variance in the B-spline estimation of the quantile functions and (clr of)

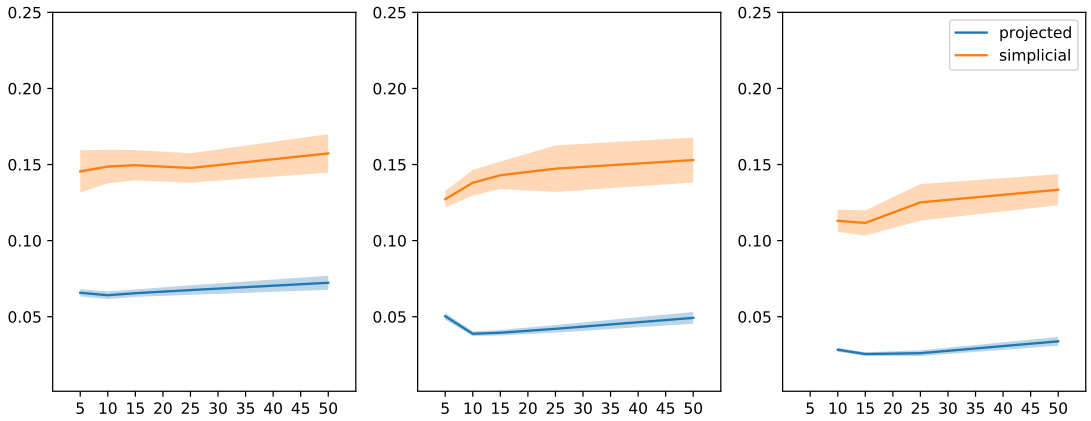



**Figure 6.12:** *Results for the third scenario. All the panels show the reconstruction error as a function of the number of the spline basis functions. From left to right the results are obtained using the* 2, 5 *and* 10 *dimensional PCA. The solid lines represent the mean of 10 independent runs on independent data sets from* (6.26) *and the shaded area represent* $\pm$ *one standard deviation.*

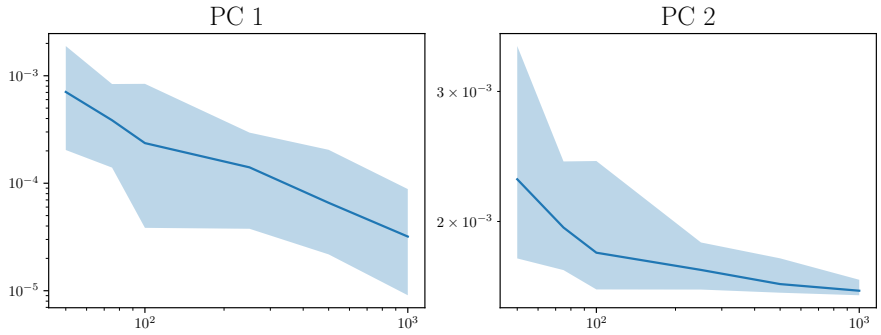
pdfs. In fact, computing the spline approximation for a single function amounts to solving a linear regression problem and increasing the dimension of the B-spline basis corresponds to increasing the number of regressors. Hence, letting $B$ the matrix with columns $\psi_1, \dots, \psi_J$ (evaluated on a grid), the variance of the OLS estimate of the coefficients $\boldsymbol{a}$ is proportional to $(B^T B)^{-1}$. When increasing the number of B-splines, the entries in $B^T B$ become closer to zero, since the support of each of the spline basis becomes smaller. This leads to smaller precision (and higher variance) in the estimator for $\boldsymbol{a}$.

Another interesting thing to notice is that the simplicial PCA exhibits a much larger variance in the reconstruction error. This is possibly due to the different degree of smoothness of the quantile functions and of the pdfs. As the quantile functions are smoother than the pdfs, their B-spline basis expansion should have lower variance and be more similar to the true quantiles.

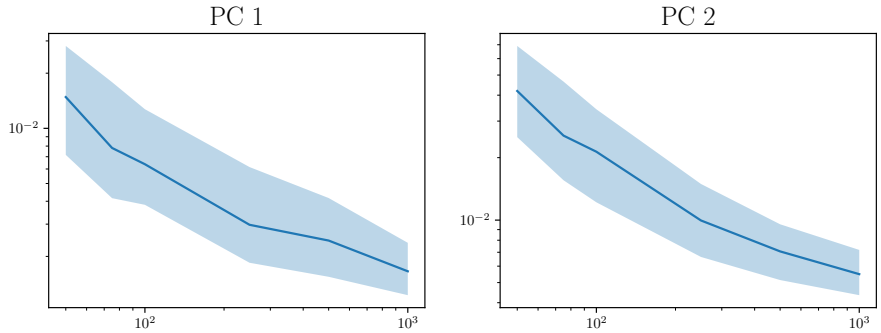
### 6.13.2 Empirical Verification of Consistency Results and Choosing $J$

In this section, we provide additional simulations to verify the consistency results established in Section 6.6.

For the PCA, we consider the two data generating processes in equations (6.19) (Gaussian) and (6.21) (DPM). First, first we fix $J = 20$ spline basis (as we do throughout Section 6.7) and let $n$ increase. Then, we also let $J$ increase linearly with $n$. We estimate the "true" principal directions by simulating $10^5$ observations and using $2500$ elements in the B-spline basis. Then, for any choice of $n$ and $J$ we generate another data set




**(a)** *Gaussian data*



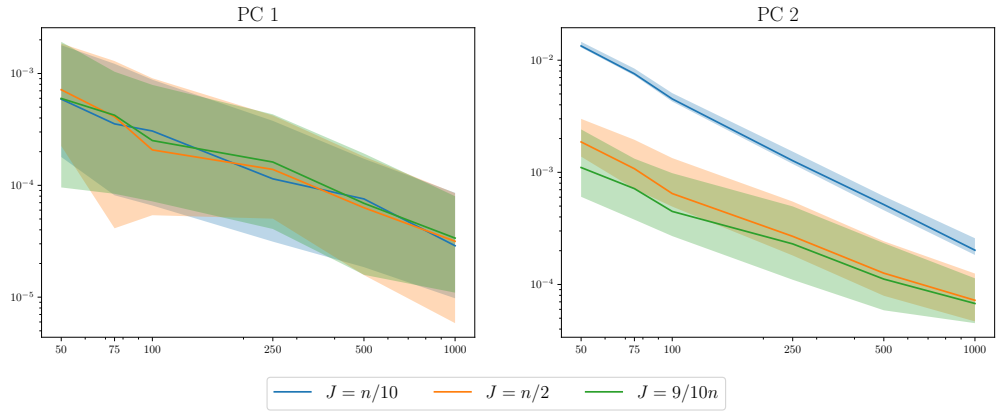

**(b)** *DPM data*

**Figure 6.13:** *$L_2$ distance between estimated and true principal directions when $J = 20$ as a function of $n$. Solid line represents the median and the shaded area to a $90\%$ confidence interval estimated from 100 independent repetition.*

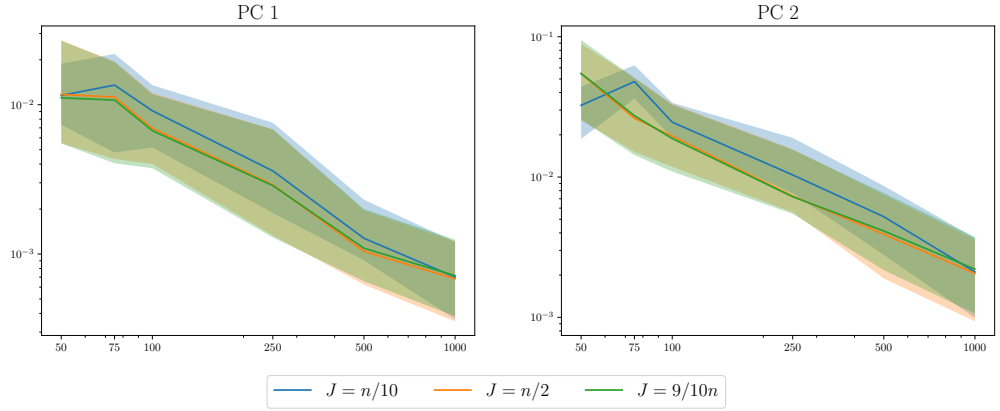
and compute the corresponding first two principal directions via the projected PCA and compute the $L_2$ norm between the "true" directions and the estimated ones.

Figure 6.13 shows the case of fixed $J$ for both data generation strategies. It is clear that in both cases the error quickly decreases to zero (observe that both the $x$ and $y$ axes are in log scale), but the convergence speed is surely sub-exponential when looking, for instance, at the second principal direction.

When increasing the number of basis elements with $n$, we consider three strategies letting $J = n/10$, $n/2$ and $9/10n$ respectively (rounded to the closest integer). Figure 6.14 shows the errors between the true and estimated principal directions in this case. Note that the convergence rate looks exponential for both data generating processes for every choice of $J = J(n)$ (increasing with $n$). In the case of Gaussian data, we observe smaller errors (as low as $10^{-5}$ for the first direction and $10^{-4}$ for the second direction) than in the case of the more challenging DPM data set, see Figure 6.14. For the former data set, using a large number of basis functions such as $9/10n$ or $n/2$ provides a much better fit than using $n/10$ basis functions on the second principal direction. For DPM data, the errors are in general two orders of magnitude higher than with




**(a)** *Gaussian data*




**(b)** *DPM data*

**Figure 6.14:** *$L_2$ distance between estimated and true principal directions as a function of $n$ for different choices of $J$. Solid line represents the median and the shaded area to a 90% confidence interval estimated from 100 independent repetition.*

Gaussian data. This is likely due to the different data generating process, which results in a more challenging problem. Interestingly, the errors are almost equal for all values of $J$ (when fixing $n$).

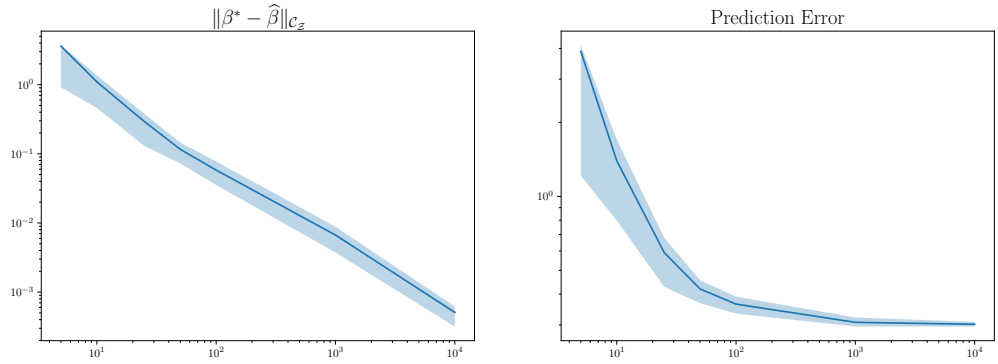
Let us now analyze the projected regression. The independent variable are generated similarly to Section 6.8, by discretizing the interval [0, 1] in 1,000 equispaced intervals, the value of the quantile function $F_{z_i}^-$ in the $j$-th interval equals $\sum_{k=1}^{j} \delta_{ik}$ and $(\delta_{i1}, \ldots, \delta_{i1000}) \sim \text{Dirichlet}(0.01, \ldots, 0.01) + \mathcal{U}([0, 5])$. We fix the kernel $\beta^\star(t, s)$ (details are given below) and let quantile functions $F^{yi} = \Pi_{L_2([0,1]^\uparrow)} \circ \Gamma_{\beta^\star}(F_{zi}^-) + \mathcal{N}(0, (0.1)^2)$.

We consider two different choices of $\beta^\star$: a smooth function $\beta_1^\star(t, s) = (t - 1/2)^3 + (s - 1/2)^3$, for which we expect that a small number of spline basis will give a low error,

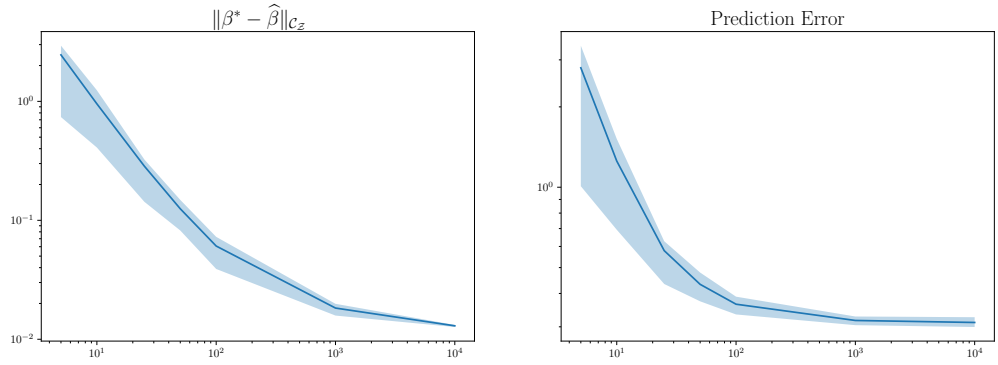
and a rougher function $\beta_2^\star(t,s)$ defined as

$$\beta_2^\star(t,s) = \sum_{k,h=1}^{10} \beta_1^\star(0.1k, 0.1h)\mathbb{I}[(t,s) \in [0.1(k-1), 0.1k) \times [0.1(h-1), 0.1h)]$$

that is, $\beta_2^\star$ corresponds to an approximation of $\beta_1^\star$ on a $10 \times 10$ grid. As in the case of PCA, we present two simulations for each choice of $\beta_i^\star$, i=1,2, where we first fix the number of spline basis $J = 20$ while increasing the sample size $n$ and second compare the performance for various values of $J$. We do not adopt the same strategy of setting $J$ as a fraction of the number of $n$ since the number of parameters to estimates grows quadratically with $J$ which makes the computational cost substantial when $J \geqslant 100$. We measure both the seminorm error $\|\widehat{\beta} - \beta^\star\|_{\mathcal{C}_Z}$ and the mean square prediction error on an unseen "test" set of $1,000$ samples.

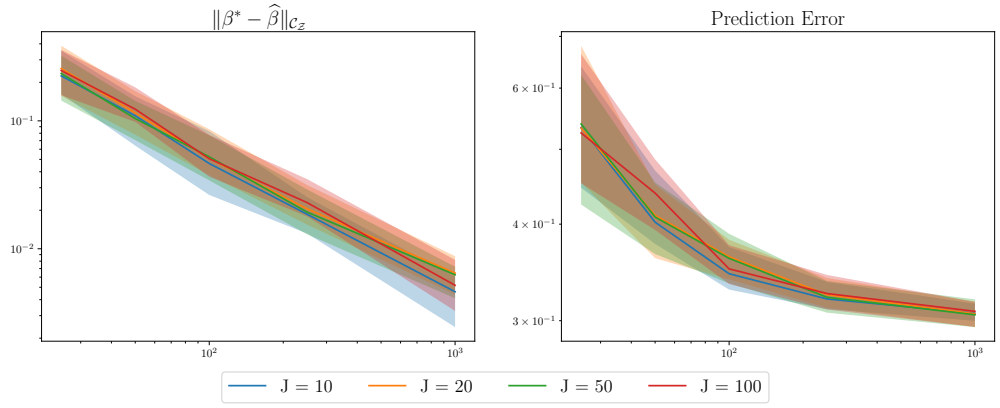



**(a)** $\beta_1^\star$

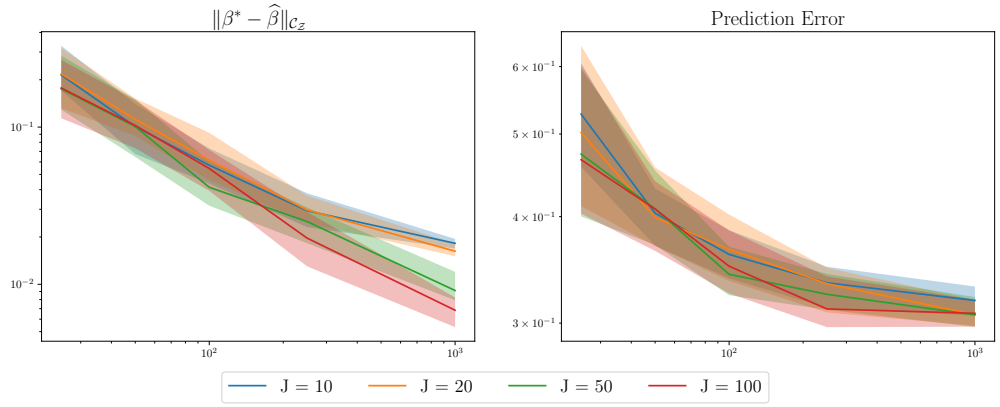



**(b)** $\beta_2^\star$

**Figure 6.15:** *Seminorm error (left) and mean square prediction error (right) for different choices of the kernel used to generate data, when $J = 20$ as a function of $n$. Solid line represents the median and the shaded area to a $90\%$ confidence interval estimated from 100 independent repetition.*




**(a)** $\beta_1^\star$




**(b)** $\beta_2^\star$

**Figure 6.16:** *Seminorm error (left) and mean square prediction error (right) for different choices of the kernel used to generate data, as a function of $n$ for different values of $J$. Solid line represents the median and the shaded area to a $90\%$ confidence interval estimated from 100 independent repetition.*

Figure 6.15 shows the seminorm error and the prediction error when $J = 20$ as $n$ increases, while in Figure 6.16 various values of $J$ are also considered. When data are generated from $\beta_1^\star$, $J = 20$ spline basis is more than enough (and actually $J = 10$ would suffice) and the seminorm error in Figure 6.15(a) and Figure 6.16(a) decays exponentially while the prediction error reaches the irreducible error with $n = 10^3$ samples. When data are generated from $\beta_2^\star$ the seminorm error does not show the same exponential decay when $J = 20$ (see Figure 6.15(b)), but it does for larger values of $J$, in particular it seems that the error obtained with $J = 50$ is the same obtained when $J = 100$, see Figure 6.16(b). Hence, it is clear that the choice of $J$ is crucial to obtain a fast decay of the error: when the kernel to be approximated is not very smooth, a larger values of spline basis elements are needed, as one would expect.

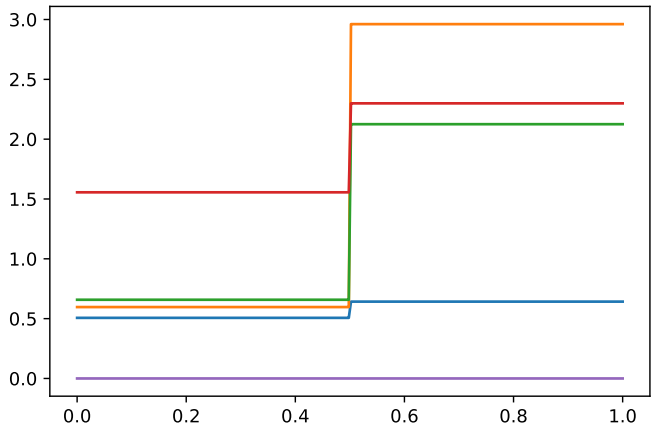



**Figure 6.17:** *Five quantile functions from the data generating process considered in Appendix 6.14.1*

We conclude this discussion by giving a practical advice on how to select $J$ for a given data set. Our suggestion is to let $J$ to be the smallest value that allows for a reconstruction error smaller than a given threshold, which may depend on the specific inferential task. For instance, if the problem is PCA and the goal is to provide a descriptive analysis of the variability, a (relative) approximation error below $0.05$ will typically give satisfactory results. If instead the goal is only to perform dimensionality reduction and working on the scores of a PCA as Euclidean data, one should aim for a lower approximation error, possibly of the order of $10^{-4}$. A similar reasoning can be applied to the regression: if the goal is mainly to interpret the estimate $\widehat{\beta}$ a larger reconstruction error can be allowed. If instead one is interested in obtaining very accurate predictions, a lower error is preferred. For instance, when $\beta_1^\star$ is used to generate the data, the reconstruction error for both dependent and independent variables is below $10^{-4}$ for $J \geqslant 20$, while to get to the same error when $\beta_2^\star$ is used one must use $J = 100$ basis.

## 6.14 Limitations of the projected framework

### 6.14.1 When the projected PCA performs poorly

Here, we show an example to highlight the limitations of the proposed framework, specifically of the projected PCA. The main idea behind this example is that the projected principal directions will be different from the nested geodesic ones when data are concentrated around the "borders" of $X$, as in the trivial example shown in Figure 6.1. In the Wasserstein case, $X$ is the space of quantile functions so that the border composed of functions that are constant on a subset of $[0, 1]$.

Hence, we consider the following data generating process, modeling directly the

quantile functions

$$F_i^-(t) = \begin{cases} v_{i1}, & \text{if } t < 0.5 \\ v_{i1} + v_{i2}, & \text{if } t > 0.5 \end{cases}$$

where $v_{ij} \sim \max\{0, \mathcal{N}(0,1)\}$ independently. See Figure 6.17 for a random sample from this data generating process.

In this case, computing the projected PCA results in an interpretability score $IS_k$ equal to one for $k = 1, 2$ and equal to zero for $k = 3, 4, \ldots$. Hence, from the third principal direction onward, the projected PCA does not give any reliable information and, if those directions are needed, in this case a nested PCA could be preferred. Despite the poor interpretability scores from the third direction onward, the reconstruction errors are always good as $NRE_1 = 0.26$ and $NRE_k \approx 10^{-6}$ for $k \geqslant 2$. Moreover, the ghost variances $GV_k$ are smaller than $10^{-10}$ for all values of $k$, so that this particular data set would be a good candidate for the hybrid methods mentioned in Section 6.7.2.

In summary, in our experience, the performance of the projected PCA can suffer when considering the interpretability of the directions associated to lower variability, but usually (at least always in our examples) gives a reasonable reconstruction error and ghost variance.

### 6.14.2 Inconsistent scores when increasing dimensions

Here, we highlight a feature which is shared by both projected and nested PCA, that is, the scores of the projection onto a projected principal component are dependent on the dimension of the principal component, as already noted in Section 6.3.1.

This can be considered a limitation to those frameworks, because it contributes to the complexity of the analysis: one has always to fix the dimension of the chosen principal component and use the scores accordingly obtained. For instance, the scores, both for nested and projected PCAs, coincide with the $L_2$ scores when the dimension of the principal components is equal to the cardinality of the spline basis $J$. This happens because the principal components are not linear subspaces. As a consequence also the interpretability score of a direction is dimension-dependent.

Hence, the choice of the dimension $k$ must be carried out balancing (i) a parsimonious representation, (ii) a low reconstruction error, so that the projections on the principal components yield good approximations of the data, and (iii) the intepretability score of the directions.

Thus, opposed to standard Euclidean PCA, where the $k+1$-th direction does not change the behavior of the data along the previous $k$ directions (i.e., the scores), when doing (any) PCA in Wasserstein space the whole picture must always be taken into account, both for nested and projected PCA to assess the interpretability of the results.

Finally, note that such interpretability might be low for both intrinsic and extrinsic methods, but this means that the Wasserstein metric may not be the most adequate to capture and explain the variability of the data set.

CHAPTER 7

# Conclusion

Throughout the chapters of this dissertation we explore two different areas in which contaminations between geometry and data analysis are, if not mandatory, at least very beneficial for the analysis: functional data analysis (up to parametrization) and distributional data analysis. To contribute to such fields we differentiate our works between two directions: summarizing informations by means of topological representations well defined across equivalence classes and extending vector calculus and related tools for spaces with possibly weird structures.

Following the manuscript, the reader immediately realizes that, even if those directions speak very different languages, they complement each other in a very fruitful way: any time a new representation of some data is given, there is a need for mathematical tools in the space of those representations. In the same fashion, we develop a class of computable statistical tools for the Wasserstein space, based on the geometry of a particular representation of its elements. This same representation greatly improves the computational costs of already existing methods. Thus, the results obtained in the thesis display that both the theoretic and the computational aspects of the problems considered can benefit from the geometric perspective pursued in the dissertation. All the novelties introduced present some kinds of drawbacks: for instance the *projected* methods introduced in Chapter 6 may suffer interpretability issues when the variability of the data is too high; the usage of dendrograms (Chapter 2, Chapter 3 and Chapter 4) instead is limited by the metric's computational cost and by the complexity of the metric space they live in. At the end of each chapter we go more in depth of all such issues as well as

proposing further developments and new research directions that can enrich the topics contained in the chapters. Moreover, Chapter 5 collects a detailed description of a set of possible developments which relate jointly to Chapter 2, Chapter 3 and Chapter 4. Here, instead, we want to conclude the manuscript with a brief, more general consideration, which links the fields of topological data analysis and Wasserstein metrics.

A general drawback of TDA's approach, is the difficulty in transferring information from the space of representations back to the original data set: interpreting results not just in terms of topological features but in terms of more specific features of the initial data is a challenging problem. The main reasons for this complexity are the invariance properties of TDA's tools: the operator which maps a datum into its topological representation is highly non-injective and thus is not clear how one can try to "invert" this map. There are however works which suggest possible paths that can be followed to do so (see for instance Gameiro et al. (2016)): induce modifications on base objects, based on modifications of the associated topological summaries. In this way, for instance, a geodesic between merge trees, induces a geodesic between functions, and so we can read the variability explained by the topological information, directly in the functions space. This point of view creates potential bridges with the Wasserstein metrics for distributions in $\mathbb{R}^n$, which have often been used to match point clouds and shapes (Liu et al., 2019; Shi et al., 2016; Solomon, 2018). Thus, one could use topological summaries to induce local deformations in objects by means of Optimal Transport and work with such deformations.

This potential research direction is an appropriate conclusion of this dissertation, since it connects the two main areas we discussed, and proposes a further contribution to the framework described in the thesis.

CHAPTER 8

# Code

All the examples and case studies involved in this thesis have been developed using original libraries coded in Python. Code has been collected into two separated packages: a package for dealing with dendrograms and one for dealing with univariate distributions with the Wasserstein metric. While the second package is already finished and readily avaiable on github, the first one is currently under development.

## 8.1 Dendrograms

In this Section we present the main files and functions contained in the package developed for dendrograms, along with the notebooks which containt the code to replicate the analises presented in previous chapters.

The core of the package is contained in the the following files:

- *Trees_OPT.py*: contain the definition of class *Tree*, which enables the creation of a dendrogram object starting from an height function and a list of vertices encoding the merging structure of a fixed basis filtration; several useful methods are implemented as well;

- *Utils_OPT.py*: contains many utilities which are needed to work with graphs and preprocess functions;
- *Utils_dendrograms_OPT.py*: contains utilities to create dendrograms starting from different kinds of data: namely functions on grids, functions on triangulations and hierarchical dendrograms obtained via other Python libraries implementing herarchical clustering;
- *top_TED_lineare_multiplicity.py*: implements the tree edit distance for dendrograms with multiplicity with values in $\mathbb{R}^n$, which can be numbers, "proper" multivariate vectors and discretized functions.

### 8.1.1 *Trees_OPT.py*

The class *Tree* needs three objects to be defined:

- *f_uniq*: is the vector of critical values $t_i$.
- *plt_tree* which is a ordered list of the vertices which appear in the tree structure. We consider the connected component as indexed by their order of appearance. For each vertex we record a triplet containing the following three numbers: the index of the associated critical value, then we have the couple of connected components merging, expressed in decreasing order. With $(i, m, -1)$ we record that the $i$-th point in the tree structure is a leaf, associated to the birth of the $m$-th component.
- *name_vertices*: it's list containing the names of the vertices. This is fundamental to work with specific subtrees of a tree.

The weighted tree structure is then encoded via a weight matrix (called *weights*), with positive weights at the coordinates of connected vertices with the weight being the weight value of the edge in the associated merge tree. The edge $[i, j]$ is then represented by $weights[i, j] = w_T([i, j])$.

We report here also the most important methods implemented for this class:

- there are a number of methods which are employed to work with the tree structures, such as: *make_edges*, *find_father*, *find_children*, ...
- the paths to the root are fundamental for the calculation of $d_E$ and thus are calculated and written in the variable *paths*:
- *sub_tree* is a method devoted to extracting a subtree of a certain vertex

- for visualization purposes, the merge tree can be written into newick format (Huerta-Cepas et al., 2010) with the function *make_newick* and then drawn with the function *plot_newick* which uses the library biopython (Chapman and Chang, 2000) to visualize the merge tree;
- the function *make_mult* takes care of the multiplicity function, which is encoded with a dictionary and can be created in different ways. The values $d(\varphi(e), 0)$ are calculated with *make_norms_mult*.

### 8.1.2 *Utils_OPT.py*

The only function which is relevant for the purposes of this thesis is *preprocess_f* which is used when extracting the merge tree of a function which is not injective or is very close to not being injective.

### 8.1.3 *Utils_dendrograms_OPT.py*

There are several utilities in this file, which are used to extract dendrograms from data and to prune them. These are the two main pipelines:

- when we want to extract a merge tree representing the clustering structure of a point cloud we use the library scipy (Bressert, 2012) inside the function *dendrolink* to extract the hierarchical clustering according to some selected linkage. Then we translate the output of the library scipy in terms of *plot_tree* and *f_uniq*, with the function *Z_to_plt_Tree*. The function *dendrolink* then return a *Tree* object.
- when we have a triangulated domain or a point cloud and we want to extract a dendrogram from a scalar function, we must call the function *from_cloud_to_dendro_sublvl*. The most important operations inside this function are taken care of by another function, *sublvl_set_filtration_multiplicity*. If the domain is not triangulated, this second function first builds a naive graph structure connecting all the points whose distance is below some threshold, and assumes that the intrinsic dimension of the domain is $1$. Then it proceeds by calculating the evolution of the connected components of such graph, or of the given triangulated domain along with a multiplicity function. The multiplicity function employed is the one calculating the measure of the sublevel sets as explained in Chapter 2. Of course multiplicity function can be redefined in a second stage.

Lastly there are the functions which are employed to prune dendrograms. Namely:

- *prune_vertices* is the function which, given a dendrogram, prunes all the leaves whose values $d(\varphi(e), 0)$ is below a threshold.
- *prune_dendro*: recursively calls *prune_vertices* until there are no more leaves to be pruned. It implements the pruning operator $\mathcal{P}_\varepsilon$.

- *prune_dendro_N*: given a certain number of leaves, this function keeps on recursively pruning the dendrogram, with increasing thresholds, until the number of leaves does not exceed the given value.

### 8.1.4 *top_TED_lineare_multiplicity.py*

This file containt the code to calculate metric $d_E$ between two *Tree* objects with multiplicities. The function *top_TED_lineare* implements the algorithm described in Section 2.7, and is quite complicated, but we try to describe it without going too much into details.

The main functions called by *top_TED_lineare* are the following:

- *make_sub_trees*: prepares all the subtrees of the given dendrograms along with the names of their vertices and the cost of their deletion
- *make_W* is the function where most of the calculations are carried out; it is the function which implements the "for" loop in the algorithm to calculate the matrix *W*. Going through the levels in the dendrograms, for every pair of points $(x, y)$ in the for loop, the function *calculate_Wxy* (or the equivalent parallelizable function) is a wrapper for the function *make_model*, which implements the optimization problem in Equation (2.4). The outputs of *make_model* and *calculate_Wxy* are the number $d_E(T_x, T_y)$ and a minimizing mapping.

At the end, *top_TED_lineare* returns either just the value $d_E(T, T')$ or this value along with a minimizing mapping.

Now we take a closer look to the function *make_model* to see how the implementation of Equation (2.4) is carried out. The package employed to write the ILP part is Hart et al. (2017), which turns out to be handy because, one the model is prepared, it can be fed into any installed and compatible solver. In the function *make_model* the following happens:

- we create the model variable, *cost*, and instantiate the optimization variables $\delta_{i,j}^{v,w}$ as in Equation (2.4), contained in an array called $L$;
- we prepare a function, *objective_poly*, which is the function to be optimized. This function is a wrapper for *make_poly* which in turns wraps around *sym_objective_fun*, where the functions $F^C$, $F^D$, $F^-$ and $F^S$ are calculated using pyomo variables. The function *eval_mapping* is equivalent to *sym_objective_fun*, but instead works with vectors and can be used to evaluate the cost function for a specific $\delta$.
- after the cost function is added to the model, the function *make_constraints* adds also the constaints.
- the optimization problem is then solved (via a chosen solver) and, if required, the minimizing mapping, along with the cost of each edit, is then extracted from the solution.

### 8.1.5 Jupyter Notebooks

We briefly list the notebooks specifying the examples they implement:

- *Dendrogram_vs_PD.ipynb*: implements the example in Section 2.8.3
- *Dendro_multiplicity_functions.ipynb*: implements the example in Section 2.8.3
- *Simulation_study_1.ipynb*: contains the simulation study in Section 3.8.1
- *Simulation_study_2.ipynb*: contains the simulation study in Section 3.8.2
- *Aneurisk_notebook.ipynb*: contains the analyses used in the case study in Section 3.9.

## 8.2 Projected Methods in 1-D Wasserstein Spaces

The repository `https://github.com/mberaha/ProjectedWasserstein` concerning the Projected Wasserstein methods is readily avaiable and much easier to navigate in. The code has been cleaned and the methods employed have a more straightforward implementation based on the formulas provided in the thesis. Thus we only give a very high level description of the files containing the main classes and methods necessary to run our examples.

- *distributions.py* contains the *Distribution* class, which is a representation of probability distributions on the real line with easily accessible quantile function, pdf and cdf. Can be initialized both with pdf and cdf. Along with that, we defined also functions to work within the simplicial framework.
- *spline.py* implements a versatile class of splines, with readily avaiable metric structure of the vector space they induce. Along with that we have the class *MonotoneQuadraticSplineBasis* which gives a monotonic spline expansion obtained with a metric projection on the cone of monotone splines.

In the folder *dimsensionality_reduction* there are our implementations of the different PCAs used in Chapter 6, namely:

- *geodesic_pca.py* implements the global geodesic PCA as in Section 6.5.3, using pyomo and the solver ipopt
- *nested_pca.py* implements the nested geodesic PCA as in Section 6.5.3, using pyomo and the solver ipopt
- *projected_pca_distrib.py* contains the class to obtain a projected PCA

- *simplicial_pca.py* contains the class to make PCA in the Bayes Space

In the *regression* folder, instead, there are different scripts implementing different kind of regression techniques. Some of them have been used in Chapter 6 and some others have not:

- *distrib_on_distrib.py* gives the distribution on distribution projected linear regression, which is described in Section 6.3.2
- *logistic.py* contains an implementation of a logistic regression for distributional data, with data living in the Wasserstein space. The model is not described in the thesis, but is well commented in the script and its definition is quite natural.
- *multi_distrib_on_distrib.py* implements a multivariate distribution on distribution regression, which has been used in Section 6.8.2.
- *scalar_on_function.py* implements scalar on distribution regression, for distribution living in the Wasserstein space
- *simplicial.py* implements distribution on distribution regression between Bayes Spaces, as described in Section 6.12.

Data and Jupyter notebooks to replicate the examples and the analyses can be easily recognised in the online repository:

- *Compare Projections.ipynb* compare the PAVA algorithm and the metric projection with splines (Section 6.4.3)
- *Comparison Regression.ipynb* can be used to replicate results in the comparison between simplicial and projected distribution on distribution regression (Section 6.8)
- *Comparison vs Simplicial PCA.ipynb* contains the code to run the comparison between the simplicial and the Wasserstein PCAS (Section 6.7)
- *Covid Deaths.ipynb* runs the analysis made with PCA on Covid data (Section 6.7)
- *Wind Forecast.ipynb* runs the multivariate distribution on distribution regression example found in Section 6.8.

# Bibliography

H. Adams, T. Emerson, M. Kirby, R. Neville, C. Peterson, P. Shipman, S. Chepushtanova, E. Hanson, F. Motta, and L. Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *J. Mach. Learn. Res.*, 18:8:1–8:35, 2017.

S.-i. Amari. Information geometry. *Japanese Journal of Mathematics*, 16(1):1–48, 2021.

L. Ambrosio, N. Gigli, and G. Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2008.

D. Anevski and P. Soulier. Monotone spectral density estimation. *The Annals of Statistics*, 39(1):418–438, 2011.

D. Anevski, O. Hössjer, et al. A general asymptotic scheme for inference under order restrictions. *The Annals of Statistics*, 34(4):1874–1930, 2006.

V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Log-euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 56(2):411–421, 2006.

J.-P. Aubin and H. Frankowska. *Set-valued analysis*. Springer Science & Business Media, 2009.

M. Audin, M. Damian, and R. Erné. *Morse theory and Floer homology*. Springer, 2014.

N. Ay, P. Gibilisco, and F. Matus. *Information Geometry and Its Applications*. Springer, 2018.

M. Ayer, H. D. Brunk, G. M. Ewing, W. T. Reid, and E. Silverman. An empirical distribution function for sampling with incomplete information. *The Annals of Mathematical Statistics*, 26(4):641–647, 1955. ISSN 00034851.

M. Banerjee, R. Chakraborty, E. Ofori, D. Vaillancourt, and B. C. Vemuri. Nonlinear regression on riemannian manifolds and its applications to neuro-image analysis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 719–727. Springer, 2015.

F. Bassetti, A. Bodini, and E. Regazzini. On minimum kantorovich distance estimators. *Statistics & probability letters*, 76(12):1298–1302, 2006.

U. Bauer, X. Ge, and Y. Wang. Measuring distance between reeb graphs. In *Proceedings of the thirtieth annual symposium on Computational geometry*, pages 464–473, 2014a.

U. Bauer, E. Munch, and Y. Wang. Strong equivalence of the interleaving and functional distortion metrics for reeb graphs. *arXiv preprint arXiv:1412.6646*, 2014b.

U. Bauer, B. Di Fabio, and C. Landi. An edit distance for reeb graphs. 2016.

U. Bauer, C. Landi, and F. Mémoli. The reeb graph edit distance is universal. *Foundations of Computational Mathematics*, pages 1–24, 2020.

K. Beketayev, D. Yeliussizov, D. Morozov, G. H. Weber, and B. Hamann. Measuring the distance between merge trees. In *Topological Methods in Data Analysis and Visualization*, 2014.

S. Berezin and A. Miftakhov. On barycenters of probability measures. *arXiv preprint arXiv:1911.07680*, 2019.

M. G. Bergomi, P. Frosini, D. Giorgi, and N. Quercioli. Towards a topological–geometrical theory of group equivariant non-expansive operators for data analysis and machine learning. *Nature Machine Intelligence*, 1(9):423–433, 2019.

E. Bernton, P. E. Jacob, M. Gerber, and C. P. Robert. On parameter estimation with the wasserstein distance. *Information and Inference: A Journal of the IMA*, 8(4): 657–676, 2019a.

E. Bernton, P. E. Jacob, M. Gerber, C. P. Robert, et al. Approximate bayesian computation with the wasserstein distance. *Journal of the Royal Statistical Society Series B*, 81(2):235–269, 2019b.

M. J. Best and N. Chakravarti. Active set algorithms for isotonic regression; a unifying framework. *Mathematical Programming*, 47(1-3):425–439, 1990.

R. N. Bhattacharya, L. Ellingson, X. Liu, V. Patrangenaru, and M. Crane. Extrinsic analysis on manifolds is computationally faster than intrinsic analysis with applications to quality control by machine vision. *Applied Stochastic Models in Business and Industry*, 28(3):222–235, 2012.

S. Bhattacharya, R. Ghrist, and V. Kumar. Persistent homology for path planning in uncertain environments. *IEEE Transactions on Robotics*, 31:1–13, 06 2015. doi: 10.1109/TRO.2015.2412051.

S. Biasotti, D. Giorgi, M. Spagnuolo, and B. Falcidieno. Reeb graphs for shape analysis and applications. *Theoretical Computer Science*, 392:5–22, 02 2008. doi: 10.1016/j.tcs.2007.10.018.

J. Bigot, R. Gouet, T. Klein, A. López, et al. Geodesic PCA in the Wasserstein space by convex PCA. In *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, volume 53, pages 1–26. Institut Henri Poincaré, 2017.

P. Bille. A survey on tree edit distance and related problems. *Theoretical computer science*, 337(1-3):217–239, 2005.

L. J. Billera, S. P. Holmes, and K. Vogtmann. Geometry of the space of phylogenetic trees. *Advances in Applied Mathematics*, 27(4):733–767, 2001.

A. Bock, H. Doraiswamy, A. Summers, and C. Silva. Topoangler: Interactive topology-based extraction of fishes. *IEEE Transactions on Visualization and Computer Graphics*, PP:1–1, 08 2017. doi: 10.1109/TVCG.2017.2743980.

A. Bône. *Learning adapted coordinate systems for the statistical analysis of anatomical shapes. Applications to Alzheimer's disease progression modeling.* PhD thesis, Sorbonne Université, 2020.

F. Borceux. *Handbook of categorical algebra: volume 1, Basic category theory*, volume 1. Cambridge University Press, 1994.

P. Breiding, S. Kališnik, B. Sturmfels, and M. Weinstein. Learning algebraic varieties from samples. *Revista Matemática Complutense*, 31(3):545–593, 2018.

E. Bressert. Scipy and numpy: an overview for developers. 2012.

M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

P. Bubenik. Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research*, 16(3):77–102, 2015.

M. Budninskiy, G. Yin, L. Feng, Y. Tong, and M. Desbrun. Parallel transport unfolding: a connection-based manifold learning approach. *SIAM Journal on Applied Algebra and Geometry*, 3(2):266–291, 2019.

T. T. Cai and P. Hall. Prediction in functional linear regression. *The Annals of Statistics*, 34:2159–2179, 2006.

A. Calissano, A. Feragen, and S. Vantini. Populations of unlabeled networks: Graph space geometry and geodesic principal components. 2020.

J. Cao, L. Mo, Y. Zhang, K. Jia, C. Shen, and M. Tan. Multi-marginal Wasserstein GAN. In *Advances in Neural Information Processing Systems*, pages 1776–1786, 2019.

G. Carlsson, T. Ishkhanov, V. De Silva, and A. Zomorodian. On the local behavior of spaces of natural images. *International journal of computer vision*, 76(1):1–12, 2008.

M. Carrière and S. Oudot. Local equivalence and intrinsic metrics between reeb graphs. *arXiv preprint arXiv:1703.02901*, 2017.

M. Catalano, A. Lijoi, and I. Prünster. Measuring dependence in the wasserstein distance for bayesian nonparametric models. *The Annals of Statistics*, forthcoming, 2021.

E. Cazelles, V. Seguy, J. Bigot, M. Cuturi, and N. Papadakis. Geodesic PCA versus log-PCA of histograms in the Wasserstein space. *SIAM Journal on Scientific Computing*, 40(2):B429–B456, 2018.

B. Chapman and J. Chang. Biopython: Python tools for computational biology. *ACM Sigbio Newsletter*, 20(2):15–19, 2000.

F. Chazal and B. Michel. An introduction to topological data analysis: fundamental and practical aspects for data scientists. *arXiv preprint arXiv:1710.04019*, 2017.

F. Chazal, B. T. Fasy, F. Lecci, A. Rinaldo, and L. Wasserman. Stochastic convergence of persistence landscapes and silhouettes. *J. Comput. Geom.*, 6:140–161, 2015.

F. Chazal, V. De Silva, M. Glisse, and S. Oudot. *The structure and stability of persistence modules*. Springer, 2016.

Y. Chen, Z. Lin, and H.-G. Müller. Wasserstein regression. *arXiv preprint arXiv:2006.09660*, 2020.

M. K. Chung, P. Bubenik, and P. T. Kim. Persistence diagrams of cortical surface data. In J. L. Prince, D. L. Pham, and K. J. Myers, editors, *Information Processing in Medical Imaging*, pages 386–397, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-02498-6.

D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37:103–120, 2007.

D. Cohen-Steiner, H. Edelsbrunner, J. Harer, and Y. Mileyko. Lipschitz functions have lp-stable persistence. *Foundations of Computational Mathematics*, 10:127–139, 02 2010. doi: 10.1007/s10208-010-9060-6.

J. M. Curry. *Sheaves, cosheaves and applications*. University of Pennsylvania, 2014.

C. Curto. What can topology tell us about the neural code? *Bulletin of the American Mathematical Society*, 54, 05 2016. doi: 10.1090/bull/1554.

M. Cuturi. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In *Advances in Neural Information Processing Systems*, pages 2292–2300, 2013.

M. Cuturi and A. Doucet. Fast Computation of Wasserstein Barycenters. In *International Conference on Machine Learning*, pages 685–693, 2014.

M. Cuturi, O. Teboul, and J.-P. Vert. Differentiable Ranking and Sorting using Optimal Transport. In *Advances in Neural Information Processing Systems*, pages 6861–6871, 2019.

P. Das and S. Ghosal. Bayesian quantile regression using random B-spline series prior. *Computational Statistics & Data Analysis*, 109:121–143, 2017.

J. Dauxois, A. Pousse, and Y. Romain. Asymptotic Theory for the Principal Component Analysis of a Vector Random Function: Some Applications to Statistical Inference. *Journal of Multivariate Analysis*, 12:136–154, 1982.

B. C. Davis. Medical image analysis via fréchet means of diffeomorphisms. 2008.

C. De Boor and J. W. Daniel. Splines with Nonnegative B-spline Coefficients. *Mathematics of computation*, 28(126):565–568, 1974a.

C. De Boor and J. W. Daniel. Splines with Nonnegative B-spline Coefficients. *Mathematics of computation*, 28(126):565–568, 1974b.

V. De Silva, E. Munch, and A. Patel. Categorified reeb graphs. *Discrete & Computational Geometry*, 55(4):854–906, 2016.

P. Delicado. Dimensionality reduction when data are density functions. *Computational Statistics & Data Analysis*, 55:401–420, 01 2011.

F. Deutsch. *Best Approximation in Inner-Product Spaces*. Springer Science & Business Media, 2012.

B. Di Fabio and C. Landi. Reeb graphs of curves are stable under function perturbations. *Mathematical Methods in the Applied Sciences*, 35(12):1456–1471, 2012.

B. Di Fabio and C. Landi. The edit distance for reeb graphs of surfaces. *Discrete & Computational Geometry*, 55(2):423–461, 2016.

I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis*. Wiley, Chichester, 1998.

R. Dykstra, T. Robertson, and F. T. Wright. *Advances in Order Restricted Statistical Inference: Proceedings of the Symposium on Order Restricted Statistical Inference Held in Iowa City, Iowa, September 11–13, 1985*, volume 37. Springer Science & Business Media, 2012.

M. L. Eaton. Group invariance applications in statistics. In *Regional conference series in Probability and Statistics*, pages i–133. JSTOR, 1989.

H. Edelsbrunner and J. Harer. Persistent homology-a survey. *Contemporary mathematics*, 453:257–282, 2008.

H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28:511–533, 2002.

J. J. Egozcue, J. L. Díaz-Barrero, and V. Pawlowsky-Glahn. Hilbert Space of Probability Density Functions Based on Aitchison Geometry. *Acta Mathematica Sinica*, 22(4): 1175–1182, 2006.

Y. Elkin and V. Kurlin. The mergegram of a dendrogram and its stability. *ArXiv*, abs/2007.11278, 2020.

K. Emmett, B. Schweinhart, and R. Rabadan. Multiscale topology of chromatin folding. In *BICT*, 2015.

B. Fasy, F. Lecci, A. Rinaldo, L. Wasserman, S. Balakrishnan, and A. Singh. Confidence sets for persistence diagrams. *The Annals of Statistics*, 42:2301–2339, 03 2014.

J. Felsenstein and J. Felenstein. *Inferring phylogenies*, volume 2. Sinauer associates Sunderland, MA, 2004.

A. Feragen, P. Lo, M. de Bruijne, M. Nielsen, and F. Lauze. Toward a theory of statistical tree-shape analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12 2012. doi: 10.1109/TPAMI.2012.265.

T. S. Ferguson. Bayesian density estimation by mixtures of normal distributions. In *Recent advances in statistics*, pages 287–302. Elsevier, 1983.

F. Ferraty and P. Vieu. *Nonparametric functional data analysis: theory and practice*. Springer Science & Business Media, 2006.

P. Fletcher. Geodesic Regression and the Theory of Least Squares on Riemannian Manifolds. *International Journal of Computer Vision*, 105, 11 2013.

P. T. Fletcher, C. Lu, S. M. Pizer, and S. Joshi. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE transactions on medical imaging*, 23(8): 995–1005, 2004.

M. Gameiro, Y. Hiraoka, S. Izumi, M. Kramár, K. Mischaikow, and V. Nanda. A topological measurement of protein compressibility. *Japan Journal of Industrial and Applied Mathematics*, 32:1–17, 03 2014. doi: 10.1007/s13160-014-0153-5.

M. Gameiro, Y. Hiraoka, and I. Obayashi. Continuation of point clouds via persistence diagrams. *Physica D: Nonlinear Phenomena*, 334:118–132, 2016.

M. K. Garba, T. M. Nye, J. Lueg, and S. F. Huckemann. Information geometry for phylogenetic trees. *Journal of Mathematical Biology*, 82(3):1–39, 2021.

E. Gasparovic, E. Munch, S. Oudot, K. Turner, B. Wang, and Y. Wang. Intrinsic interleaving distance for merge trees. *ArXiv*, abs/1908.00063, 2019.

C. Giusti, R. Ghrist, and D. Bassett. Two's company, three (or more) is a simplex: Algebraic-topological tools for understanding higher-order structure in neural data. *Journal of Computational Neuroscience*, 41, 01 2016. doi: 10.1007/s10827-016-0608-6.

W. E. Hart, C. D. Laird, J.-P. Watson, D. L. Woodruff, G. A. Hackebeil, B. L. Nicholson, J. D. Siirola, et al. *Pyomo-optimization modeling in python*, volume 67. Springer, 2017.

A. Hatcher. *Algebraic topology*. Cambridge Univ. Press, Cambridge, 2000.

J. Hein, T. Jiang, L. Wang, and K. Zhang. On the complexity of comparing evolutionary trees. In Z. Galil and E. Ukkonen, editors, *Combinatorial Pattern Matching*, pages 177–190, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg. ISBN 978-3-540-49412-6.

C. Hofer, R. Kwitt, M. Niethammer, and A. Uhl. Deep learning with topological signatures. In *NIPS*, 2017.

E. Hong, Y. Kobayashi, and A. Yamamoto. Improved methods for computing distances between unordered trees using integer programming. In *International Conference on Combinatorial Optimization and Applications*, pages 45–60. Springer, 2017.

K. Hron, A. Menafoglio, M. Templ, K. Hrůzová, and P. Filzmoser. Simplicial principal component analysis for density functions in Bayes spaces. *Computational Statistics & Data Analysis*, 94:330–350, 07 2014.

S. Huckemann, T. Hotz, and A. Munk. Intrinsic shape analysis: Geodesic pca for riemannian manifolds modulo isometric lie group actions. *Statistica Sinica*, pages 1–58, 2010a.

S. Huckemann, T. Hotzand, and A. Munk. Intrinsic shape analysis: Geodesic PCA for Riemannian manifolds modulo isometric lie group actions. *Statistica Sinica*, 20:1–58, 2010b.

S. F. Huckemann and B. Eltzner. Backward nested descriptors asymptotics with inference on stem cell differentiation. *The Annals of Statistics*, 46(5):1994–2019, 2018.

J. Huerta-Cepas, J. Dopazo, and T. Gabaldón. Ete: a python environment for tree exploration. *BMC bioinformatics*, 11(1):1–7, 2010.

I. M. James. *The topology of Stiefel manifolds*, volume 24. Cambridge University Press, 1976.

S. Jung, I. L. Dryden, and J. S. Marron. Analysis of principal nested spheres. *Biometrika*, 99(3):551–568, 2012.

H. Karcher. Riemannian center of mass and mollifier smoothing. *Communications on pure and applied mathematics*, 30(5):509–541, 1977.

A. Kneip and K. J. Utikal. Inference for Density Families Using Functional Principal Component Analysis. *Journal of the American Statistical Association*, 96(454):519–542, 2001.

J. Koperwas and K. Walczak. Tree edit distance for leaf-labelled trees on free leafset and its comparison with frequent subsplit dissimilarity and popular distance measures. *BMC bioinformatics*, 12:204, 05 2011. doi: 10.1186/1471-2105-12-204.

V. Kovacev-Nikolic, P. Bubenik, D. Nikolic, and G. Heo. Using persistent homology and dynamical distances to analyze protein binding. *Statistical applications in genetics and molecular biology*, 15:19–38, 03 2016. doi: 10.1515/sagmb-2015-0057.

M. Kramár, A. Goullet, L. Kondic, and K. Mischaikow. Persistence of force networks in compressed granular media. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 87:042207, 04 2013. doi: 10.1103/PhysRevE.87.042207.

D. Krupka and D. Saunders. *Handbook of global analysis*. Elsevier, 2011.

V. Kumar, Y. Gu, S. Basu, A. Berglund, S. A. Eschrich, M. B. Schabath, K. Forster, H. J. Aerts, A. Dekker, D. Fenstermacher, et al. Radiomics: the process and the challenges. *Magnetic resonance imaging*, 30(9):1234–1248, 2012.

T. Lacombe, M. Cuturi, and S. Oudot. Large scale computation of means and clusters for persistence diagrams using optimal transport. In *NeurIPS*, 2018.

B. Lavine and J. Workman. Chemometrics. *Analytical chemistry*, 80(12):4519–4531, 2008.

G. Lawton. Men hit harder by covid-19. *New Scientist*, 246(3279):8, 2020. ISSN 0262-4079. doi: https://doi.org/10.1016/S0262-4079(20)30786-7.

J. Le-Rademacher and L. Billard. Principal component analysis for histogram-valued data. *Advances in Data Analysis and Classification*, 11(2):327–351, 2017.

J.-G. Liu, R. L. Pego, and D. Slepčev. Least action principles for incompressible flows and geodesics between shapes. *Calculus of Variations and Partial Differential Equations*, 58(5):1–43, 2019.

P. Lum, G. Singh, A. Lehman, T. Ishkanov, M. Alagappan, J. Carlsson, G. Carlsson, and M. Vejdemo Johansson. Extracting insights from the shape of complex data using topology. *Scientific Reports*, 3, Feb. 2013. ISSN 2045-2322. doi: 10.1038/srep01236.

R. MacPherson and B. Schweinhart. Measuring shape with topology. *Journal of Mathematical Physics*, 53, 11 2010. doi: 10.1063/1.4737391.

S. Maletić, Y. Zhao, and M. Rajkovic. Persistent topological features of dynamical systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 26, 10 2015. doi: 10.1063/1.4949472.

J. S. Marron, J. Ramsay, L. Sangalli, and A. Srivastava. Statistics of time warpings and phase variations. *Electronic Journal of Statistics*, 8:1697–1702, 2014.

J. S. Marron, J. Ramsay, L. Sangalli, and A. Srivastava. Functional data analysis of amplitude and phase variation. *Statistical Science*, 30(4):468–484, 2015.

A. Menafoglio, A. Guadagnini, and P. Secchi. A kriging approach based on Aitchison geometry for the characterization of particle-size curves in heterogeneous aquifers. *Stochastic Environmental Research and Risk Assessment*, 28(7):1835–1851, 2014.

P. Michor, D. Mumford, J. Shah, and L. Younes. A metric on shape space with explicit geodesics. *Atti Accad. Naz. Lincei Cl. Sci. Fis. Mat. Natur. Rend. Lincei (9) Mat. Appl.*, 19, 07 2007. doi: 10.4171/RLM/506.

Y. Mileyko, S. Mukherjee, and J. Harer. Probability measures on the space of persistence diagrams. *Inverse Problems - INVERSE PROBL*, 27, 12 2011. doi: 10.1088/0266-5611/27/12/124007.

M. Moakher and M. Zéraï. The riemannian geometry of the space of positive-definite matrices and its application to the regularization of positive-definite matrix-valued data. *Journal of Mathematical Imaging and Vision*, 40(2):171–187, 2011.

G. Monge. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris*, 1781.

D. Morozov and G. Weber. Distributed merge trees. In *PPoPP '13*, 2013.

D. Morozov, K. Beketayev, and G. Weber. Interleaving distance between merge trees. *Discrete and Computational Geometry*, 49:22–45, 01 2013.

J. R. Munkres. *Elements of algebraic topology*. CRC press, 2018.

F. Murtagh and P. Contreras. Algorithms for hierarchical clustering: An overview, ii. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7, 09 2017. doi: 10.1002/widm.1219.

P. Nagabhushan and R. Pradeep Kumar. Histogram PCA. In D. Liu, S. Fei, Z. Hou, H. Zhang, and C. Sun, editors, *Advances in Neural Networks – ISNN 2007*, pages 1012–1021, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

G. Naitzat, A. Zhitnikov, and L.-H. Lim. Topology of deep neural networks. *Journal of Machine Learning Research*, 21(184):1–40, 2020.

S. Pal, T. J. Moore, R. Ramanathan, and A. Swami. Comparative topological signatures of growing collaboration networks. In B. Gonçalves, R. Menezes, R. Sinatra, and V. Zlatic, editors, *Complex Networks VIII*, pages 201–209, Cham, 2017. Springer International Publishing. ISBN 978-3-319-54241-6.

V. M. Panaretos and Y. Zemel. *An Invitation to Statistics in Wasserstein Space*. Springer Nature, 2020.

V. Pascucci and K. Cole-McLaughlin. Parallel computation of the topology of level sets. *Algorithmica*, 38:249–268, 10 2003. doi: 10.1007/s00453-003-1052-3.

V. Patrangenaru and L. Ellingson. *Nonparametric Statistics on Manifolds and Their Application to Object Data Analysis*. CRC Press, 2015.

V. Pawlowsky-Glahn, J. J. Egozcue, and K. Van den Boogaart. Bayes hilbert spaces. *Australian &amp New Zealand Journal of Statistics*, 56:171–194, 06 2014.

M. Pegoraro. A metric for tree-like topological summaries, 2021.

M. Pegoraro and M. Beraha. Projected statistical methods for distributional data on the real line with the wasserstein metric, 2021.

M. Pegoraro and P. Secchi. Functional data representation with merge trees, 2021.

X. Pennec. Intrinsic Statistics on Riemannian Manifolds: Basic Tools for Geometric Measurements. *Journal of Mathematical Imaging and Vision*, 25:127–154, 07 2006.

X. Pennec. Statistical Computing on Manifolds: From Riemannian geometry to Computational Anatomy. In *LIX Fall Colloquium on Emerging Trends in Visual Computing*, pages 347–386. Springer, 2008.

X. Pennec. Barycentric subspace analysis on manifolds. *The Annals of Statistics*, 46 (6A):2711–2746, 2018.

X. Pennec, S. Sommer, and T. Fletcher. *Riemannian Geometric Statistics in Medical Image Analysis*. Academic Press, 2019.

J. Perea and J. Harer. Sliding windows and persistence: An application of topological methods to signal analysis. *Foundations of Computational Mathematics*, 15, 07 2013. doi: 10.1007/s10208-014-9206-z.

J. Perea, A. Deckard, S. Haase, and J. Harer. Sw1pers: Sliding windows and 1-persistence scoring; discovering periodicity in gene expression time series data. *BMC bioinformatics*, 16:257, 08 2015. doi: 10.1186/s12859-015-0645-6.

G. Peyré, M. Cuturi, et al. Computational Optimal Transport: With Applications to Data Science. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019.

D. Pigoli, J. A. Aston, I. L. Dryden, and P. Secchi. Distances and inference for covariance operators. *Biometrika*, 101(2):409–422, 2014.

F. Pokorny, M. Hawasly, and S. Ramamoorthy. Topological trajectory classification with filtrations of simplicial complexes and persistent homology. *The International Journal of Robotics Research*, 35, 08 2015. doi: 10.1177/0278364915586713.

F. A. Potra and S. J. Wright. Interior-point methods. *Journal of Computational and Applied Mathematics*, 124(1-2):281–302, 2000.

S. Potter, M. Del Negro, G. Topa, and W. Van der Klaauw. The advantages of probabilistic survey questions. *Review of Economic Analysis*, 9(1):1–32, 2017.

L. Prchal and P. Sarda. Spline estimator for functional linear regression with functional response. *Technical Report*, 2007.

N. Pya and S. N. Wood. Shape constrained additive models. *Statistics and Computing*, 25(3):543–559, 2015.

X. Qi and H. Zhao. Some theoretical properties of Silverman's method for smoothed functional principal component analysis. *Journal of Multivariate Analysis*, 102:741–767, 2011.

J. O. Ramsay. Functional data analysis. *Encyclopedia of Statistical Sciences*, 4, 2004.

J. O. Ramsay and B. W. Silverman. *Functional Data Analysis*. Springer, New York, NY, USA, 2005.

B. Ripley and U. Grenander. General pattern theory: A mathematical theory of regular structures. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 158:635, 01 1995. doi: 10.2307/2983457.

A. Rizvi, P. Camara, E. Kandror, T. Roberts, I. Schieren, T. Maniatis, and R. Rabadan. Single-cell topological rna-seq analysis reveals insights into cellular differentiation and development. *Nature Biotechnology*, 35, 05 2017. doi: 10.1038/nbt.3854.

R. Rockafellar and R. J.-B. Wets. *Variational Analysis*. Springer Verlag, Heidelberg, Berlin, New York, 1998.

O. Rodríguez, E. Diday, and S. Winsberg. Generalization of the Principal Components Analysis to Histogram Data. pages 12–16, 2000.

L. Sangalli, P. Secchi, S. Vantini, and A. Veneziani. Efficient estimation of three-dimensional curves and their derivatives by free-knot regression splines, applied to the analysis of inner carotid artery centrelines. *Journal of the Royal Statistical Society Series C*, 58:285–306, 07 2009a. doi: 10.1111/j.1467-9876.2008.00653.x.

L. Sangalli, P. Secchi, S. Vantini, and V. Vitelli. K-mean alignment for curve clustering. *Computational Statistics & Data Analysis*, 54:1219–1233, 05 2010. doi: 10.1016/j.csda.2009.12.008.

L. Sangalli, P. Secchi, and S. Vantini. Analysis of aneurisk65 data: $k$-mean alignment. *Electronic Journal of Statistics*, 8:1891–1904, 2014.

L. M. Sangalli, P. Secchi, S. Vantini, and A. Veneziani. A case study in exploratory functional data analysis: Geometrical features of the internal carotid artery. *Journal of the American Statistical Association*, 104(485):37–48, 2009b.

K. Sekitani and Y. Yamamoto. A recursive algorithm for finding the minimum norm point in a polytope and a pair of closest points in two polytopes. *Mathematical Programming*, 61:233–249, 1993.

J. Shi, W. Zhang, and Y. Wang. Shape analysis with hyperbolic wasserstein distance. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5051–5061, 2016.

Y. Shinagawa, T. L. Kunii, and Y. L. Kergosien. Surface coding based on morse theory. *IEEE Computer Graphics and Applications*, 11(5):66–78, 1991.

V. Silva and R. Ghrist. Coverage in sensor networks via persistent homology. *Algebraic & Geometric Topology*, 7, 04 2007. doi: 10.2140/agt.2007.7.339.

B. W. Silverman et al. Smoothed functional principal components analysis by choice of norm. *The Annals of Statistics*, 24(1):1–24, 1996.

A. Sizemore, C. Giusti, and D. Bassett. Classification of weighted networks through mesoscale homological features. *Journal of Complex Networks*, 5, 12 2015. doi: 10.1093/comnet/cnw013.

J. Solomon. Optimal transport on discrete domains. *AMS Short Course on Discrete Differential Geometry*, 2018.

R. Sridharamurthy, T. B. Masood, A. Kamakshidasan, and V. Natarajan. Edit distance between merge trees. *IEEE Transactions on Visualization and Computer Graphics*, 26(3):1518–1531, 2020.

A. Srivastava, I. Jermyn, and S. H. Joshi. Riemannian analysis of probability density functions with applications in vision. *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

A. Srivastava, E. Klassen, S. Joshi, and I. Jermyn. Shape analysis of elastic curves in euclidean spaces. *IEEE transactions on pattern analysis and machine intelligence*, 09 2010. doi: 10.1109/TPAMI.2010.184.

A. Srivastava, W. Wu, S. Kurtek, E. Klassen, and J. S. Marron. Registration of functional data using fisher-rao metric. *arXiv: Statistics Theory*, 2011.

S. Srivastava, V. Cevher, Q. Dinh, and D. Dunson. Wasp: Scalable Bayes via barycenters of subset posteriors. In *Artificial Intelligence and Statistics*, pages 912–920, 2015.

A. Stefanou. Tree decomposition of reeb graphs, parametrized complexity, and applications to phylogenetics. *Journal of Applied and Computational Topology*, 4(2): 281–308, 2020.

K.-C. Tai. The tree-to-tree correction problem. *J. ACM*, 26:422–433, July 1979. ISSN 0004-5411.

S. J. Taylor and B. Letham. Forecasting at scale. *The American Statistician*, 72(1): 37–45, 2018.

E. F. Touli. Frechet-like distances between two merge trees. *ArXiv*, abs/2004.10747, 2020.

E. F. Touli and Y. Wang. Fpt-algorithms for computing gromov-hausdorff and interleaving distances between trees. In *ESA*, 2018.

C. J. Tralie. *Geometric Multimedia Time Series*. PhD thesis, Duke University, 2017.

P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa. Statistical computations on grassmann and stiefel manifolds for image and video-based recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2273–2286, 2011.

K. Turner, Y. Mileyko, S. Mukherjee, and J. Harer. Frechet means for distributions of persistence diagrams. *Discrete & Computational Geometry*, 52, 06 2012. doi: 10.1007/s00454-014-9604-7.

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

S. Vantini. On the definition of phase and amplitude variability in functional data analysis. *Test*, 21:1–21, 01 2009. doi: 10.1007/s11749-011-0268-9.

R. Verde, A. Irpino, and A. Balzanella. Dimension reduction techniques for distributional symbolic data. *IEEE transactions on cybernetics*, 46, 01 2015.

C. Villani. *Optimal Transport: old and new*, volume 338. Springer Science & Business Media, 2008.

A. Waechter and L. Biegler. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106:25–56, 2006.

H. Wang and J. S. Marron. Object oriented data analysis: Sets of trees. *Ann. Statist.*, 35 (5):1849–1873, 10 2007. doi: 10.1214/009053607000000217.

Y. Wang, H. Ombao, and M. Chung. Topological data analysis of single-trial electroencephalographic signals. *The Annals of Applied Statistics*, 12:1506–1534, 09 2018. doi: 10.1214/17-AOAS1119.

K. wu and S. Zhang. A contour tree based visualization for exploring data with uncertainty. *International Journal for Uncertainty Quantification*, 3:203–223, 01 2013. doi: 10.1615/Int.J.UncertaintyQuantification.2012003956.

K. Xia, X. Feng, Y. Tong, and G. We. Persistent homology for the quantitative prediction of fullerene stability. *Journal of computational chemistry*, 36, 03 2015. doi: 10.1002/jcc.23816.

K. Xia, Z. Li, and L. Mu. Multiscale persistent functions for biomolecular structure characterization. *Bulletin of Mathematical Biology*, 80, 12 2016. doi: 10.1007/s11538-017-0362-6.

D. Xu and Y. Tian. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2, 08 2015. doi: 10.1007/s40745-015-0040-1.

Q. Yu, X. Lu, and J. S. Marron. Principal nested spheres for time-warped functional data analysis. *Journal of Computational and Graphical Statistics*, 26:144 – 151, 2013.

C. Zhang, P. Kokoszka, and A. Petersen. Wasserstein autoregressive models for density time series. *arXiv preprint arXiv:2006.12640*, 2020.

A. Zomorodian and G. Carlsson. Computing persistent homology. *Discrete and Computational Geometry*, 33:249–274, 02 2005.